DOCUMENT RESUME

ED 188 146 CS 005 512

AUTHOR Goldstein, Ira
TITLE Research on a Frame-Based Model of Reading Comprehension. Final Report.
INSTITUTION Massachusetts Inst. of Tech., Cambridge. Artificial Intelligence Lab.
SPONS AGENCY National Inst. of Education (DHEW), Washington, D.C. Basic Skills Group.
PUB DATE Aug 79
GRANT NIE-G-77-0017
NOTE 99p.: A number of pages may be marginally legible.

EDRS PRICE MF01/PC04 Plus Postage.
DESCRIPTORS *Artificial Intelligence: *Cognitive Processes: Computers: *Language Processing: *Learning Theories: Models: *Reading Comprehension: *Reading Research
IDENTIFIERS *Frame Theory

ABSTRACT

This report summarizes computational investigations of language comprehension based on Marvin Minsky's theory of frames, a recent advance in artifical intelligence theories about the representation of knowledge. The investigations discussed explored frame theory as a basis for text comprehension by implementing models of the theory and developing methods for exploiting its power: specifically they: (1) investigated how common-sense knowledge about the world can be embedded within a frame system, how that knowledge can be made "active" to model the intentions of a person planning a simple event, and the extent to which knowledge embedded in one particular frame system can be used to understand a grammatical construction called "noun, noun modification": (2) implemented a testbed system that understands connected discourses using knowledge embedded in frames to "glue together" the various pieces of information conveyed by individual sentences: and (3) studied how natural language text of very simple sorts can be immediately analyzed into the meaning of that text by exploiting the structure of individual frames. The first chapter of the report outlines frame theory, discusses briefly how this theory is embedded within Frame Representation Language (FRL--a programing language), and summarizes the major results of the investigations. The remaining three chapters discuss the results in detail. (FL)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

U S DEPARTMENT OF HEALTH,
EDUCATION & WELFARE
NATIONAL INSTITUTE OF
EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

ED188146

Final Report
Research on a Frame-Based Model of Reading Comprehension

Basic Skills Group
National Institute of Education

NIE-G-77-0017

Ira Goldstein, Principal Investigator
Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, Mass.

August 1979

CS005512

Chapter 1

Introduction

The report summarizes a computational exploration of language comprehension based upon Marvin Minsky's *theory of frames*, a recent advance in artificial intelligence theories about the representation of knowledge. Frame theory contends that (1) intelligence arises from the application of large amounts of highly specific knowledge, as opposed to a few general inferencing mechanisms, and (2) this is accomplished through the use of a library of *frames*, packets of knowledge that provide descriptions of typical objects and events. Each description contains both an abstract template providing a skeleton for describing any instance of that class of entities and a set of defaults for typical members of the class. The defaults allow the information system to supply missing detail, maintain expectations, and notice anomalies.

We have investigated the strengths and weaknesses of frame theory as a basis for text comprehension by implementing models of this theory and developing methods for exploiting its power. In particular, we have

-investigated how common sense knowledge about the world can be embedded within a frame system.

-investigated, as a precursor of dialogue understanding, how that knowledge can be made "active" to model the intentions of a person planning a simple event.

-implemented a testbed system which understands connected discourses, using knowledge embedded in frames to "glue together" the various pieces of information conveyed by individual sentences. This testbed has been used to test the interaction of this knowledge from single sentences with

JUL 15 1980

knowledge not only about past context, but also will common sense knowledge about the world.

-studied how natural language text of very simple sorts can be immediately analyzed into the "meaning" of that text by exploiting the structure of individual frames, avoiding the phase of grammatical analysis that many models of natural language analysis by computer consider obligatory.

-investigated to what extent the knowledge imbedded in one particular frame system can be used to understand a grammatical construction called "noun-noun modification", i.e. such word strings as "soup pot cover handle", where the relation between the nouns must be derived entirely from knowledge about the world with no grammatical clues.

This introduction will sketch an outline of frame theory, briefly discuss how this theory is embedded within FRL, and then summarize the major results of our investigations. The remaining chapters of this report reproduce major papers which discuss these results in detail.

A sketch of Minsky's frame theory

(The following quote is taking from Minsky's original paper on frame theory [Minsky 75]:)

When one encounters a new situation (or makes a substantial change in one's view of a problem), one selects from memory a structure called a *frame*. This is a remembered framework to be adapted to fit reality by changing details as necessary.

A *frame* is a data-structure for representing a stereotyped situation like being in a certain kind of living room or going to a child's birthday party. Attached to each frame are several kinds of information. Some of this information is about how to use the frame. Some is about what one can expect to happen next. Some is about what to do if these expectations are not confirmed.

We can think of a frame as a network of nodes and relations. The "top levels" of a frame are fixed, and represent things that are always true about the supposed situation. The lower levels have many *terminals* - "slots" that must be filled by specific instances or data. Each terminal can specify

conditions its assignments must meet. (The assignments are usually smaller "subframes".) Simple conditions are specified by markers that might require a terminal assignment to be a person, an object of sufficient value or a pointer to a subframe of a certain type. More complex conditions can specify relations among the things assigned to several terminals.

The present research was done within the context of an implementation of frame theory called FRL, a programming language embedded in the LISP programming language which provides a convenient way of constructing a frame-based knowledge system. In FRL, the subparts of a frame are called *slots*. Each slot can have attached to it *comments*, e.g. who filled that slot, *default values*, what should fill that slot if no specific filler has been mentioned, and *constraints*, which specify properties which fillers of that slot must fulfill. These constraints are arbitrary predicates written in the programming language LISP; if the "constraint" programs attached to a slot do not "complain" when some particular object is put into a slot, then that filler is an acceptable possibility for that slot of that frame. These frames are organized into a hierarchical classification system which enables the inheritance of properties. Each frame has one slot, the AKO (A Kind Of) slot, whose value is one or more other frames. Each frame is considered an "instance" of the frames in its AKO slot and inherits most of its structure from these frames. Other than the AKO slot there are no other restrictions on the number or type of slots.

Understanding Discourse Using Frames

The most important result we have achieved is the implementation of a testbed natural language understanding system called PAL that utilizes frames to understand texts of connected discourses. [This project, in the large, was jointly funded by NIE and ARPA.] Under NIE funding, we have investigated how natural language input can be added to a system that knows

about the structure of meetings,and how to schedule them. This system knows how to utilize the "focus" of a connected discourse to decide to what or whom uses of pronouns like "he", "she" "it", refer. It also knows how to use knowledge embedded in frames to abstract from each new sentence input to the system that information which is new, and which "fills out" the system's understanding of the central topic of the discourse.

A sample scenario of what PAL is designed to do is given below:

I want to schedule a meeting with Dave.
It should be at 3 p.m. on Thursday.
We can meet in his office.
Invite Bruce.

To understand this discourse, PAL must have several natural language skills: parsing for the grammatical structure of each sentence; interpretation of the logical structure; mapping of the words of each sentence to a representation used by the underlying programs; deciding to what each term like "a meeting", "Dave" and "his office" refers; and interpretation of each sentence for its purpose in the continuing discourse. All but the first two of these skills utilize the frame structures in important ways, as is discussed at length in Chapter 2 of this report.

The first two of these skills constitute the parser and case frame interpreter developed by Mitch Marcus [Marcus forthcoming]. The remainder of the system components were developed by Candy Sidner, with the assistance of Beth Levin; the resulting system was debugged and made robust by Mitch Marcus.

To present a clearer picture of what PAL must be able to do,

consider a sentence by sentence interpretation of the above dialogue.

I want to schedule a meeting with Dave.

PAL interprets an internal representation of the speaker as referent of "I," and an internal representation of "David McDonald" as the referent of "Dave."

PAL creates a new internal representation with features to be discussed later to be the referent of "a meeting."

PAL interprets "want to schedule a meeting" to be a request for a scheduling operation which may extend over several sentences.

PAL interprets the whole sentence to be asserting that the meeting has two participants, the speaker and Dave McDonald.

It should be at 3 p.m. on Thursday.

PAL interprets "it" as co-referring to the meeting under discussion.

PAL disambiguates the time phrase to a frame form used by the scheduler.

PAL interprets the sentence as asserting additional information about the meeting at hand.

We can meet in his office.

PAL determines that the speaker and other participant are the co-referent of "we."

PAL finds in its internal representations of things, an entity which "his office" can refer to.

PAL accepts the sentence as providing more information about the meeting at hand and asserts that fact.

Invite Bruce.

PAL finds an internal representation of the person referred to as "Bruce."

PAL determines that the elided event which Bruce is to attend is the meeting under discussion.

PAL accepts the invite command as asserting another participant of the meeting.

<end of discourse>

PAL interprets the scheduling request as complete and calls the scheduling system to carry out the scheduling command as it has been specified in the discourse.

Other Results

Other results of this investigation are:

We have concluded that what we call "complex nouns" phrases, like:

Mr. John P. Jones
August the 12th, 1978
Herbert Hoover Memorial High School

can be understood directly into frame structures by simply adding a small amount of syntactic information to a frame itself, circumventing most syntactic and intermediate representations. This research was done by David Dill and Dr. Mitchell Marcus. Chapter 3 reports on this research; it reproduces Mr. Dill's B.Sc. thesis.

We have looked into the possibility of using the structure of a network of frames to determine the meaning of strings of "noun-noun" modifiers like

water meter cover
flywheel adjustment screw
soup pot cover handle
reading comprehension mastery tests

Much of the information needed to understand to what such noun-noun strings refer seems to be optimally encoded with a frame system. Thus, from the fact that the "meter" frame will have a slot for "quantity metered", which "water" can fill, a system should be able to determine that a water meter is a meter which meters a quantity of water. Given that "meter" will have (say) a slot which indicates that a cover is a subpart of a meter, the system can deduce that a "water meter cover" is a cover which is part of a meter which meters a quantity of water. By attempting to implement such a system within the confines of FRL, we have discovered that for such information to be accessible, it must be in some language that can be used for inference by the underlying representation system. Because restrictions on the slots of a frame are arbitrary LISP programs which cannot be easily

examined to see what they will or will not accept, FRL seems to be inadequate in this regard. This investigation was conducted by Andrew Schulert and Mitchell Marcus. Chapter 4 reproduces Mr. Schulert's B.Sc. thesis, which discusses this issue at length.

In short, frame-based systems appear to be useful models of text comprehension. They not only facilitate the interaction of the many different sorts of knowledge need to comprehend a natural language text, but the structure of a frame itself seems to specially facilitate the understanding of various grammatical constructions in English text, which strengthens the plausibility of such systems as models of text comprehension.

References

David Dill [1979] "A Combined Syntactic and Semantic Representation for Some English Language Constructs", unpublished B.Sc. thesis, Dept. of EECS, MIT, Cambridge, Mass.

Mitch Marcus [forthcoming] *A Theory of Syntactic Recognition for Natural Language*, MIT Press, Cambridge, Mass.

Marvin Minsky [1975] "A Framework for Representing Knowledge" in The Psychology of Computer Vision, edited by Patrick Henry Winston, McGraw-Hill, New York.

Bruce Roberts & Ira Goldstein [1977] *The FRL Manual*, AI Memo 409, MIT AI Lab, Cambridge, Mass.

Andrew Schulert [1978] "On the Evaluation of Representation Systems for Semantic Knowledge", unpublished B.Sc. thesis, Dept. of EECS, MIT, Cambridge, Mass.

Candace Sidner [1978] *A Progress Report on the Discourse and Reference Components of PAL*, AI Memo 468, MIT AI Lab, Cambridge, Mass.

Chapter 2

The Discourse and Reference Components of PAL

Candace Sidner

CONTENTS

1. Introduction 3
2. Definition of Discourse 5
3. The Concept of Focus 7
4. Modules of PAL 9
5. Interpretation of Discourse Purposes 10
6. Co-reference Disambiguation 13
7. Extensions 16
8. Acknowledgements 16
9. References 16

1. Introduction

Every discourse in English consists of one or more sentences which create a general context of people, places, objects, times and actions. The speaker of the discourse generally will not relate references from one sentence to the previous in any direct fashion nor indicate how the requests or assertions of each sentence in the discourse are connected. For the hearer to interpret the speaker's discourse and decide what the speaker is requesting or asserting, the hearer must complete two tasks, among others: (1) disambiguate the referential terms for their inter-sentential and extra-sentential links, and (2) determine the purpose of each sentence in the discourse. The first of these two tasks makes it possible to know what entities the speaker is referring to. The second task results in establishing a connected discourse and understanding what the speaker wants to communicate. Interpreting the discourse purposes of various sentences explains why D1 is acceptable below (even though D1-2 does not mention the party) while D2 is unacceptable. A theory of reference disambiguation will explain the disambiguation of *his* to Bruce and not to Mike, in D3.

D1-1 John is having a party at his house.
2 I think the guest of honor is Mary as they are going to announce the publication of Mary's book.

D2-1 Henry wants to meet with Harold.
2 Sing a song before 3 on Thursday.

D3-1 I want to have a meeting this week.
2 Bruce$_i$ will be the guest lecturer.
3 He will speak on slavery in ant colonies.
4 Mike wants to read his$_i$ report before the talk.

An explanation of these phenomena underlies the research being conducted at the MIT AI lab on PAL. While PAL is designed to understand the English form of requests for arranging various events, the design depends upon a theory about how to interpret a speaker's[1] extended discourse. PAL acts as a model of a hearer in these discourse situations. Two problems that must be solved before PAL can understand requests in extended discourse are referential disambiguation and discourse purpose interpretation. This paper reports on progress on these two problems.

A sample scenario of what PAL is designed to do is given in D4 below.

D4-1 I want to schedule a meeting with Dave.
2 It should be at 3 p.m. on Thursday.
3 We can meet in his office.

---

1. I will use the term *speaker* to refer to the producer of a spoken or written discourse and *hearer* to refer to the receiver of the discourse.

4 Invite Bruce.

To understand this discourse, PAL must have several natural language skills:

a. parsing for the syntactic structure.
b. interpretation of predicate-argument relations.
c. mapping of the words of each sentence to a representation used by the underlying database and programs.
d. disambiguation of the referential terms.
e. interpretation of each sentence for its discourse purpose.

The first two of these skills constitute the parser and case frame interpreter developed by Mitch Marcus. The representation mapping was developed by the author. These three modules are discussed in Marcus [1978]. To present a clearer picture of what PAL must be able to do, consider a sentence by sentence interpretation of the above dialogue.

I want to schedule a meeting with Dave.
PAL interprets an internal representation of the speaker as referent of "I," and an internal representation of "David McDonald" as the referent of "Dave."
PAL creates a new internal representation with features to be discussed later to be the referent of "a meeting."
PAL interprets "want to schedule a meeting" to be a request for a scheduling operation which may extend over several sentences.
PAL interprets the whole sentence to be asserting that the meeting has two participants, the speaker and Dave McDonald.

It should be at 3 p.m. on Thursday.
PAL interprets "it" as co-referring to the meeting under discussion.
PAL disambiguates the time phrase to a frame form used by the scheduler.
PAL interprets the sentence as asserting additional information about the meeting at hand.

We can meet in his office.
PAL determines that the speaker and other participant are the co-referent of "we."
PAL finds in its internal representations of things, an entity which "his office" can refer to.
PAL accepts the sentence as providing more information about the meeting at hand and asserts that fact.

Invite Bruce.
PAL finds an internal representation of the person referred to as "Bruce."
PAL determines that the ellided event which Bruce is to attend is the meeting under discussion.
PAL accepts the invite command as asserting another participant of the meeting.

<end of discourse>
PAL interprets the scheduling reuqest as complete and carries out the scheduling command with the meeting as it has been specified in the discourse.

In order to perform these tasks, a theory about the nature of discourse and some of its components has been developed and will be reported on here. Following that discussion, a closer look at the rules used by an implemented running version of PAL will be discussed.

2. Definition of Discourse

First, a "discourse" must be defined. I take a discourse to be any connected piece of text or spoken language of more than one sentence or independent sentence fragment. Ideally, every discourse is about some central concept which is then elaborated by the clauses of a discourse. Speakers often produce discourses which fail to meet this specification because they talk 1a) about several concepts without relating them or 1b) without informing the hearer that several concepts will be discussed at once or 2) because there is no central concept in their discourses. However, this idealization will serve to introduce some important terms. Multi-concept discourses do occur, and can be described using an approach which is a generalized version of that presented in this paper. Some cases of multi-concept discourse are discussed in Bullwinkle [1977]. However, the theory presented here has been tested in a running implementation of PAL, and this paper is restricted to that tested model.

In previous work [Winograd, 1971; Rieger, 1973; Charniak, 1972] various structures for referencing were assumed. Winograd used lists of entities of the same semantic type and chose referents for anaphoric terms based on recency and likelihood in the proper semantic class. His mechanism was too simple and failed to account for numerous anaphoric cases as well as being limited to objects in a closed world. Rieger postulated memory structures from a conceptual dependency representation of the sentences of a discourse. The memory structures were used to infer other information that could be unified to determine co-reference. His algorithms suffer from the explosive number of inferences that can be made from each memory structure. Charniak supposed that there were large collections of inference rules, called demons, which knew what to do with a small piece of the total knowledge, and which fired whenever that knowledge was encountered. This theory represents overkill; if one could have as many demons as Charniak supposed and get them to fire when that knowledge occurred, the mechanism could be used to predict co-referentiality of referencial terms. However, controlling the multitude of demons is difficult[2], and furthermore one cannot imagine how such a collection of knowledge is learned in the first place.

To interpret definite noun phrases and anaphors, a different approach is taken in PAL. It is assumed that discourse contains a structure, which when represented, can constrain the interpretation of referential terms. From the discourse structure, rules have been discovered which govern the acceptability of referential terms in different discourse

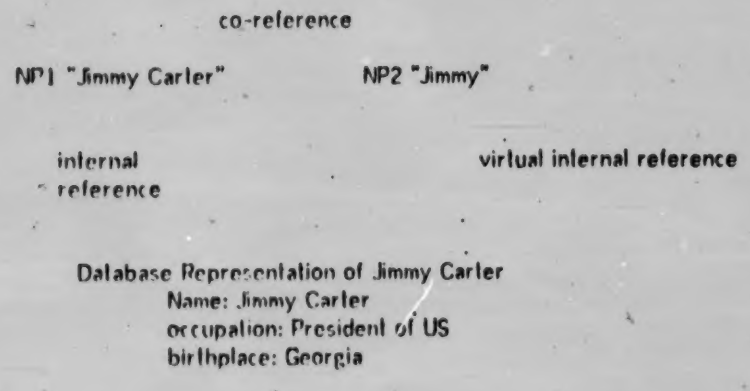
---

2. Rosenberg [personal communication] has created a device called sentinels which may partially solve this problem.

situations. The interpretation of references is not strictly deterministic; it is like knowing which of several places to look in the discourse for a co-referent and trying out the term found there.

The theory underlying PAL distinguishes two kinds of referring. The first is an internal reference between a noun phrase and some pre-existing database object. That database object represnts a real world entity. In Figure 1 below internal reference links the noun phrase NP1 "Jimmy Carter" to a representation of Jimmy Carter (who is described as president of the US, etc.). How that database object refers to the real world is the classical semantic problem of reference (cf. Kripke [1972] among others) and is beyond the scope of this work. The other kind of referring is co-reference. Co-reference links a noun phrase to another noun phrase. The two noun phrases are said to co-refer, and both refer to the same database object. In Figure 1, the dashed link from NP2 "Jimmy" to NP1 is a co-reference link. The dot-dash link from NP2 to the database object is a virtual internal reference link which results from the co-reference link from NP2 to NP1 and from the internal reference link from NP1 to the database object. Internal reference and co-reference links are distinguished because co-reference links can be established more easily using discourse structure. In the remainder of this paper when I speak of internal reference, I will drop the phrase "internal" and use only "reference."

Fig. 1. Reference Links Between Noun Phrases



3. The Concept of Focus

The central concept of a discourse may be elaborated by several sentences of the discourse and then either discontinued in favor of a related concept, or dropped in favor of a new concept. This central concept of a discourse is called the discourse focus or simply the focus. This term was first used by Grosz [1977]. A simple example of focus is *meeting* in D4 repeated below:

D4-1 I want to schedule a meeting with Dave.
2 It should be at 3 p.m. on Thursday.
3 We can meet in his office.
4 Invite Bruce.

All four sentences give information about the focussed entity. The focus is what makes a text or a set of utterances a discourse.

In this work the focus is assumed to be a concept to which other concepts are associated. Some of the association links are "built-in" in the sense that they exist previous to the discourse. For example with *meeting*, built-in association links include that a meeting has a time, a place, a set of participants, and a topic of discussion. These association links are distinguished in the sense that the concept has explicit links to these concepts while no explicit links exist to other concepts such as color, cost or age. The discourse often serves the purpose of specifying more about the concepts linked to a focus. In D4-1, there is certain information about who the participants are, while D4-2 specifies the time. D4-3 causes the hearer to infer that the office is a place for a meeting, because the focus *meeting* has a place associated with it, and because PAL expects to be informed about the concepts associated to a meeting.

In PAL the association links between concepts are easily expressed in the frames structure of FRL [Goldstein and Roberts, 1977]. A frame for a meeting has slots for times, places, participants and so on. It is exactly these slots that serve the purpose of association links to other concepts. One purpose of a discourse with PAL is to fill those slots with values and required information. As I will discuss in the section on the use of definite noun phrases, the values given to those slots are also useful in interpreting co-reference and in understanding the purpose of a sentence of the discourse.

Focus also serves as the central index point for co-referencing. The focus is what is going to be talked about in the discourse. When it is introduced, it is new information. Thereafter it is the given information, and more new information is added to it. Knowing what the focus is helps determine co-reference relations because old information can be pronominalized while new information cannot. If a focus is seen not just as an entity by itself

but connected to other entities, focus indicates how those entities can be co-referents as well. In D4 (2.4), the focus of *meeting* can be used to determine the co-reference of *it*, *we* and *his* of *his office*: *it* must co-refer to the focus, *we* to those individuals associated to the focus who include the speaker, and *his* to an individual associated to the focus who is not the speaker and has male gender. The focus is used as an access function for retrieving the co-referent of a particular noun phrase. Later in this paper, rules governing the use of anaphora by means of the focus of the discourse will be discussed.

In the current version of PAL, focus is chosen as the first noun phrase following the verb if one exists, else the subject is used as focus. This method of choosing focus is adequate for current PAL discourses but not sufficient for the most general case. See Sidner [forthcoming] for a full discussion of focus choice. Once a focus is chosen, it can be used in succeeding sentences to determine the co-reference of pronouns or definite noun phrases as well as to check to see if the discourse is still connected. A sentence like (1a) below followed by (1b) is a disconnected discourse because the co-referential terms in (1b) are unrelated to the focus of (1a) based on the association links present in the database.

(1a) I want to meet with Henry.
(1b) Give me an ice cream cone.

The focus of the discourse can be changed while maintaining a connected discourse. The chief means are end of discourse remarks and focus-shift. End of discourse remarks can be explicitly stated ones like "That's all," or implicit ones, such as the act of simply ending the input stream. A less reliable, implicit marking of the end of discourse is to use a sentence with unrelated co-referential terms. In the case above, (1a) followed by (1b) could be assumed to be two separate discourses. This case is less reliable because it is impossible to tell if the speaker assumes that the ice cream cone is related (as is often the case with a non-ideal speaker) or whether the speaker intends to change the discourse to a new one. At present PAL does not accept this kind of abrupt discourse change; instead PAL indicates that such a sentence is not intelligible in the discourse. A more sophisticated PAL might request that the speaker explain how it is that (1b) is related to the discourse.

The other means of changing the focus I call focus-shift. A discourse may expand various aspects of a focus and then choose one aspect of the focus to describe in detail. For example, in a discourse about meetings, we may want to spend several sentences specifying the time for the meeting, why that time is best and so on. When time is being discussed, one would like to know that the focus has changed so that assertions or requests can be taken to be about time. However, the meeting focus may be brought back into the discussion later. To

maintain both foci, the meeting focus is stacked for later use.[3] Detecting this focus change is the process of focus-shift.

Focus shifts cannot be predicted; they are detectable only after they occur. To detect the focus shift, the focus shift mechanism takes note of new phrases in sentences following the introductory sentence. Any new phrase is a potential focus. an anaphoric term in a sentence which follows the potential focus sentence may co-refer to either the focus or the potential focus. If the potential focus is an acceptable co-referent, it is the co-referent of the anaphoric term, and the focus shifts to the potential focus. The choice of office as co-referent of *it* in D5-3 results from focus-shift. The co-referent of *it* to meeting in D5-3' results from the rejection of the potential focus office as the co-referent.

D5-1: I want to schedule a meeting with George, Jim, Steve and Mike.
2 We can meet in my office.
3 It's kind of small, but the meeting won't last very long anyway.
3' It won't take more than 20 minutes.

Rejection of a co referent results from semantic information about the type of verb and the type of semantic entities it accepts. Semantic information has been proposed for use with co-reference (see Winograd [1971], among others). PAL uses this information only to reject or confirm choices made by the focus and focus-shift mechanisms, rather than to suggest classes of co-referents.[4]

4. Modules of PAL

The preceding description of co-reference interpretation has been incorporated into a series of modules for PAL. These modules are depicted in Figure 2 below. The arrows represent flow of control between modules.

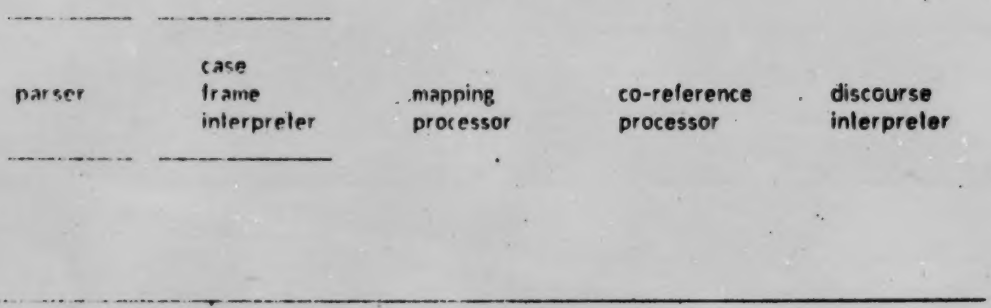
Each English sentence presented to PAL by a speaker is interpreted via a parser, case frame interpreter and representation mapping program [Bullwinkle, 1976; Marcus, 1978] into a set of FRL frames. The sentence "Schedule a meeting in my office," is represented by the following simplified frames (slot and slot values are listed also).

---

3. Grosz [Deutsch, 1975] gave the first specification of discourse shifts using the concept of focus. These are discussed further in Grosz [1977].

4. The mechanism of focus-shift is discussed in more detail in Bullwinkle [1977], where the term "sub-topic shift" is used.

Fig. 2. Modules of PAL




```
frame          =  schedule201
   a-kind-of   =  schedule
   type        =  "imperative"
   actor       =  PAL
   event       =  meeting203

frame          =  meeting203
   a-kind-of   =  meeting
   place       =  office207
   determiner  =  "a"

frame          =  office207
   a-kind of   =  office
   determiner  =  my209

frame          =  my209
   a-kind-of   =  my
```

Given these frames, PAL is expected to determine what my209, office207, and meeting203 co-refer to. PAL also must decide what the purpose of an imperative scheduling request (represented by schedule201) is relative to its database collection of actions. Each of these modules will now be discussed in detail.

5. Interpretation of Discourse Purposes

To interpret discourse purposes, a discourse module creates a model of the discourse and controls the process of focus identification. Since the beginning, middle and end of a discourse each require different actions by the PAL scheduler, the discourse component models each differently. The first sentence of the discourse is assumed to specify what the nature of the user's communication is. This is a simplified view of the real communication process. Many discourses do not simply state their object and then elaborate the relevant

information. Instead many speakers begin a discourse as in D6 below in which the first sentence contains a reason for some other action, which is requested in a later sentence. Other discourses may introduce individuals or objects to the hearer for later comment on them.

D6: I am going on vacation the beginning of next week. John wants to see me, so schedule our regular meeting session before I leave.

The current version of PAL uses the simplified view of discourse to choose a discourse purpose. Introductory sentences are assumed to be making some sort of request. The PAL discourse module chooses which request on the basis of the verb and any associated modals, or on the basis of verbs of desire (*want, wish, would like*) and the verb complement. A request consists not only of the request type, but of some object which the request is about (intransitive verbs are not relevant to PAL since telling PAL to laugh, run or groan is inappropriate). The focus of the discourse is used for this purpose. This choice is plausible not only because the focus is closely associated with the object of the verb, but also because a discourse centers discussion on some particular entity, and that entity is captured by the focus.

Once a focus has been designated, sentences occurring in mid-discourse are assumed to be about the focus until the co-reference module predicts a focus-shift and/or until the verbs used are inconsistent with the discourse request. Mid-discourse sentences often do not explicitly co-refer to the focus as has been shown previously in D1 and D4; they may contain an implicit focus co-reference. Use of focus for co-reference disambiguation has the added benefit that sentences containing implicit focus co-references are easily recognized by the discourse component. Once an implicit focus relation is established, the module can go onto predictions of focus shift. Knowledge that the speaker is co-referring to the focus, either explicitly or implicitly, makes possible the prediction that the discourse is not yet complete, and the prediction that the speaker is making a coherent request. Since neither prediction can be assumed trivially true, the focus is important to the communication process.

In addition to the focus, the discourse module contains knowledge allowing the module to decide if the verb of a new sentence is consistent with the discourse request. Thus in D7 below, the second sentence uses a verb that is consistent with the scheduling request while in D7', the verb is odd.

D7: Henry wants to meet with Harold. Choose a time before 3 on Thursday.
D7': Henry wants to meet with Harold. Sing a song before 3 on Thursday.

The knowledge needed to predict consistency is represented in the frames database in two ways. First the frame for the discourse request contains information about what other requests can be sub-requests of the discourse. Second a set of mapping frames contain

information which determine how a verb can be interpreted as making a certain request. For example, the verb *be* can be associated with scheduling and re-scheduling activities. However, the intention of the speaker in a sentence like (2) is different within the context of a scheduling or a re-scheduling request.

(2) The time should be 3 pm.

In a scheduling context, (2) can be interpreted to request that the time be established as 3 pm while (2) in re-scheduling can have an interpretation of changing the time from whatever it was to 3 pm. PAL captures the intention of the speaker relative to a request context by an inference mechanism which is a matcher that determines that (2) represented as a frame[5] can be associated with scheduling requests by a simple mapping between two frames. This correspondence coupled with the use of focus makes it possible to understand (2) as part of a discourse.

In addition, the mapping functions tell how to interpret the current sentence into one of the commands which the scheduler can perform. Included in this process are how to map the slots of one frame into a frame which is the scheduling action. For example, the verb frame for "We can meet in 823" is mapped from a "meet" frame into a frame called "assert" with a slot for the object asserted, which is the focus, and a slot for what is asserted about that object, in this case the place as 823.

The end of a discourse is currently interpreted as being the end of the speaker's input stream. A more sophisticated means of interpreting discourse end is possible, though not implemented, given the focus mechanism: when the needed slots of the focus are filled, the speaker can be considered to have finished this discourse. Upon sensing the end of the discourse, the discourse module informs the scheduler that it can carry out the action requested at the discourse beginning. At first glance this may appear as if the discourse request specified at the beginning is ignored in favor of other requests. In fact the initial request is used in interpreting mid-discourse sentences. However, many discourse actions like scheduling require that the action of scheduling be delayed until all the necessary information for scheduling is presented. This process normally cannot be stated in a single sentence, and a whole discourse is needed to fill in the request. In this fashion the discourse module reflects the fact that a discourse consists of many sub-discourses centered around individual entities and which are opened and closed by focus shifting or finishing discussion of the current focus.

---

5. A frame is not taken as the meaning, in the classical semantic sense, for (2); PAL makes no claims about this sense of meaning.

PAL is similar to the GUS system [Bobrow et al, 1977] because it expects a discourse to provide information about the slots of a frame. GUS permits user initiative although it is unclear what the extent of this initiative is. GUS does not seem to allow for user initiative of the discourse requests. Since PAL expects full user control over all parts of the discourse, PAL needs a complete description of the discourse and its focus. PAL's use of focus also presents a complete theory of the kinds of co-reference problems raised by the GUS system.

6. Co-reference Disambiguation

There are two sub-modules for co-reference interpretation in PAL, the sentential and inter-sentential co-reference modules. The inter-sentential co-reference sub-module chooses co-references for referential terms in the discourse once the focus is identified. The task of determining co-reference varies depending upon the presence or absence of previous discourse. When there is previous discourse, co-reference interpretation depends largely on the focus. For simple[6] definite noun phrases, PAL assumes either the focus is the direct co-referent of the definite noun phrase or the focus contains a slot that is the co-reference of the definite noun phrase. This assumption needs modification since some definite noun phrases are used to refer outside the context of the discourse. For example, when trying to schedule a meeting, if the speaker says (3), the definite noun phrase co-refers to an entity associated with the meeting under discussion; that association is reflected in the frame slot structure of FRL.

(3) The best place is my office.

However, if the speaker says (4), *the conference room*, i.e. that particular conference room which the speaker has in mind, is not associated with meetings in general, and so the focus does not point out the co-reference.

(4) We ought to meet in the conference room.

However, by searching the focus, the lack of a connection can be noticed, and a reference from the database can then be considered. In this way, the focus acts as an access function, but only for those co-referential terms related to the previous sentences of the discourse.

PAL uses database search with growing contexts of reference to choose reference for other kinds of noun phrases which refer to entities outside the discourse. Growing a context is accomplished using the immediate set of frames from the first sentence and recursively creating larger sets from the slot values of those frames until the frame with the

---

6. A simple definite noun phrase is a definite noun phrase containing no relative clauses. At present PAL interprets only such noun phrases.

name in question is found. The context growing mechanism reduces search from a more global search strategy, and helps control potential ambiguities that exist due to multiple possible references in the database. This same method could be used for definite noun phrases that refer outside the discourse.

Use of the focus is actually somewhat more complex since the definite noun phrase may be a co-reference to the potential focus of the discourse. Should a definite noun phrase co-refer to the potential focus, the discourse module pushes the current focus to a focus stack and takes the potential focus as the new focus. The pushed focus is available for later use in the discourse. The current inter-sentential sub-module does not interpret definite noun phrases used generically. The focus can be used for these cases as well (see Sidner, [forthcoming]), but the details of this process are not included in the current version of PAL.

The inter-sentential co-reference sub-module also determines the co-reference of personal pronouns. For the pronouns of first person plural (*we, us*), two choices can be made. First, the sub-module can choose the focus as the direct co-referent of the anaphor. Second the sub-module can choose a set of co-references from a particular slot of the focus. That slot must contain co-references including the speaker of the discourse. For *he/she*, and its object forms, the focus is chosen as a direct co-reference. Using the focus as co-referent explains the anaphoric co-reference in D8 of *his* to Bruce and rather than Mike. When the focus is not the co-referent, a co-referent stipulated by the co-reference rules of the sentential co-reference sub-module, discussed below is used. Finally if neither is acceptable, entities associated with the focus are checked for co-reference. This sub-module predicts misuse of *he/she* pronouns if no co-references are found from this process or if more than one results from the last step in the process.

The interpretation of co-reference for *he/she* pronouns needs to be expanded to include consideration of potential focus since in D8 below, *his* co-refers to Bruce and not to Mike.

D8: I want to have a meeting this week. Bruce$_i$ will be the guest lecturer. Mike wants to read his$_i$ report first.

It appears that the focus and potential focus ought to be checked for co-reference to such pronouns before sentential co-reference rules are used. However, further experimentation with such cases is needed to confirm this aspect of co-reference.

For the co-reference of *it*, the inter-sentential co-reference sub-module chooses a co-referent either from the focus, the potential focus or from predictions from sentential co-reference rules, which are discussed below. This choice strategy is not entirely adequate

because recency appears to play a role in the co-reference choices for *it*. Recency rules are discussed in Sidner [forthcoming], and could be included in a future version of PAL. The inter-sentential co-reference sub-module uses the semantic constraints placed on the pronoun by the verb in a few instances; this portion of PAL could be expanded greatly. Co-reference rules for *they* work similarly to those for *it* with consideration that the speaker cannot be included in the co-reference set.

When no previous discourse exists, PAL's sentential co-reference sub-module uses the co-reference rules of Lasnik [1976] to choose co-references. The rule is stated as follows: If a noun phrase, NP_1, precedes another noun phrase, NP_2, and NP_2 is not a pronoun, and further if the minimal cyclic node dominating NP_1 also dominates NP_2, then NP_2 and NP_2 are disjoint in reference. The expression "disjoint in reference" is taken to mean have no references in common, thereby blocking the co-reference of *Bob* and *Tom* to *they* in (5).

(5) They assume that Bob will talk to Tom.

By using Lasnik's rule, disjoint references of a noun phrase in a sentence can be chosen, as well as a list of acceptable co-references for the noun phrase. This information is recorded in the frame presenting the noun phrase. As pointed out by Reinhart [1976], Lasnik's rule fails to predict the disjoint references in sentences like (6) and (7) below, but these cases are not problematic given inter-sentential co-reference rules because other rules will predict the co-reference for the pronouns first.

(6) Near Dan, he saw a snake.
(7) For Ben's wife, he would give his life.

In addition to the use of a co-reference rule, the sentential sub-module determines the referents of proper names. Using the collection of frames which make up the discourse, a frame containing the correct first (and if given, last) name can be found. Should the immediate discourse fail to produce the name referent, a larger context can be grown from the slot values and from the slot defaults of the frame representing the focus. The same context growing mechanism used for definite noun phrases is used. By this process of context growing, ambiguous uses of names like *John* can be avoided. *John* will refer to that person most closely associated with the discourse. If more than one frame for the name John is found, the context growing process predicts that the speaker has used the name ambiguously. Context growing has been effective in a limited number of cases tested so far, although a database with more potential ambiguities would further test this sub-module.

7. Extensions

The current PAL can be expanded in many directions. Some of the necessary developments of its co-reference capabilities have already been discussed. Significantly, these capabilities do not require extensive new theoretical appparatus; the focus of discourse and structure of FRL can sustain the needed improvements. In discourse interpretation PAL must be extended to interpret discourses which define new people, places, events, actions and like objects as well as to interpret preferences of users and purposes for various activities. These extensions not only will make PAL a more useful system, but also they encompass a set of tasks useful for other interactive programming domains. Experimentation on the discourse module of PAL is need to incorporate these new capabilities.

8. Acknowledgements

The author wishes to thank Gretchen Brown, Ira Goldstein and Steve Rosenberg for their comments and suggestions on drafts of this paper.

9. References

Bobrow, D., R. Kaplan, M. Kay, D. Norman, H. Thompson, T. Winograd, [1977] *GUS, A Frame-Driven Dialogue System*, Artificial Intelligence, Volume 8, Number 2, April.

Bullwinkle, C. [1976] *The Semantic Component of PAL: the Personal Assistant Language Understanding Program*, MIT AI Laboratory Working Paper, March.

Bullwinkle, C. [1977] *Levels of Compexity in Discourse for Anaphora Disambiguation and Speech Act Interpretation*, Proceedings of the Fifth International Joint Conference in Artificial Intelligence, August 1977; also M.I.T. A.I. Lab Memo 413.

Charniak, E. [1972] *Toward a Model Of Children's Story Comprehension*, M.I.T. A.I. Lab TR-266.

Deutsch, B. [1975] *Establishing Context In Task-Oriented Dialogues*, Proceedings of the 13 Annual Meeting of ACL, AJCL Microfiche 35.

Goldstein, I. P. and R. B. Roberts. [1977] *NUDGE, A Knowledge-based Scheduling Program*, M.I.T. A.I. Lab memo 405.

Grosz, Barbara [1977] *The Representation and Use of Focus in Dialogue Understanding*. Stanford Research Institute Technical Note 151, Menlo Park, California

Kripke, Saul A. [1972] *Naming and Necessity*. in Semantics of Natural Language, Davidson and Harman (eds.) Reidel Publishing Co., Boston.

Lasnik, Howard. [1976] *Remarks on Co-reference*, Linguistic Analysis, Volume 2, Number 1.

Marcus, Mitchell [1978] *Progress Report on the Parser and Semantics of PAL*, M.I.T. A.I. Lab

memo forthcoming.

Reinhart, Tanya [1976] *The Syntactic Domain of Anaphora*, unpublished Ph.D. dissertation, Department of Foreign Literature and Linguistics, M.I.T.

Rieger, Charles J. [1974] *Conceptual Memory: A Theory and Computer Program for Processing the Meaning Content of Natural Language Utterances.* Stanford Artificial Intelligence Lab Memo AIM 233.

Sidner, C. [forthcoming] *A Computational Model of Co-reference Comprehension in English.* Ph.D. dissertation, M.I.T.

Winograd, Terry [1971] *Procedures as a Representation for Data in a Computer Program for Understanding Natural Language.* M.I.T. dissertation.

Chapter 3

A Combined Syntactic and Semantic Representation for Some English Language Constructs

David L. Dill

The grammar of many of the lower-level constituents of grammatical structure has not been an area of exciting linguistic discovery, in contrast with study of clause-level phenomena. The syntax of these *conceptual phrase*, as they will be termed here, seems to be somwehat *ad hoc*, which presents problems for their specification in natural language understanding models.

In the following chapter, Dill argues that the irregular behavior of these phrases stems from a closer relationship between the syntax and semantics of these than other English constructs; Dill argues that conceptual phrases of a given type all correspond to objects which are instances of the same underlying frame. As a result, semantic knowledge embedded in the prototypic frame can be used to "optimize" the process of communicating them and understanding them.

Dill shows how the unique nature of conceptual phrases can be exploited by building *syntactically augmented frames* that are much simpler that the sum total of separate syntactic and semantic representations would be, using the structure of frames to simplify the understanding process.

Table of Contents

1. Introduction 4
2. Observations about the Behavior of Conceptual Phrases 7
3. A Combined Syntactic and Semantic Representation 12
   - 3.1 Introduction 12
   - 3.2 Protocols for Frame Communication 12
   - 3.3 The Syntactic Description 14
   - 3.4 Relation between the Protocol and the Syntactic Description 16
   - 3.5 The Representation and Meaning 18
4. Numbers: A Sample Representation 20
   - 4.1 A Representation for Numbers 20
   - 4.2 The Validity of the Representation 25
5. When to Expect a Conceptual Phrase 28
6. Conclusions 32
   - 6.1 Where Conceptual Phrases Fit In 32
   - 6.2 directions for future research 33

I. A Description of Parsifal 35

II. A Description of FRL 42

1. Introduction

This thesis was motivated by a desire to facilitate the construction of a more complete grammar for Parsifal, a computer program that parses written English. The original program and grammar were implemented in an attempt to provide a better computational description of linguistic phenomena, relying heavily on the results of current research in transformational grammar.

Since research in generative grammar has concentrated primarily on phenomena involving the structure and manipulation of syntactic constructs at the level of detail represented by clauses and noun phrases, the main focus in the development of the current Parsifal grammar has concentrated on explaining these 'higher-level' phenomena.

Parsifal has been successful in achieving its goal of explaining linguistic observations about the nature of these higher-level constituents. It is noted in the original paper describing Parsifal [5] that its grammar for clause-level syntactic structures, though simple, is more complete than either W. Woods' LUNAR system [12], or Winograd's SHRDLU [11], since these both tend to concentrate on parsing 'phrase-level' constituents.

However, there are many other issues that must be dealt with in order to arrive at a English understanding system that is truly practical. Some of the more obvious issues are how to use information derived from discourse context, and how to represent knowledge in a form that facilitates semantic interpretation. Another issue that is highlighted by Parsifal's clause-level orientation, is the question of how to deal with the 'special-purpose' phrases that seem to occur frequently in natural language. The last issue is the basis of this thesis.

Ideally, the constraints that allow such an elegant clause-level grammar would also allow the same simplicity in the description of more detailed phrases. Unfortunately, this is not the case. The grammars for numbers, names, and dates, which are 'lower-level' in the sense that they are always constituents of noun phrases, seem to lose any sense of either elegance or theory. It should be noted that Parsifal's grammars for these constructs are not necessarily any worse than grammars for such constructs in other English parsers; they simply fail to measure up to the quality of Parsifal's grammar for higher-level constituents. The contrast between the relatively ad-hoc implementation of lower-level constituents and the highly perspicuous representation for clause-level syntax leads to speculation that a different set of constraints might result in a better grammar for the lower-level phrases.

This thesis investigates the nature of these special-purpose phrases (which I will henceforth term 'conceptual phrases') in more detail. In addition, a method of representing them is presented that takes advantage of their distinct nature. Finally, the role of the resulting system is examined in the context of current research in natural language processing.

-Chapter 2 analyzes the existing grammar for conceptual phrases, and concludes that they correspond to single semantic entities (hence the name 'conceptual phrase').

-Chapter 3 develops a succinct representation for conceptual phrases by augmenting a semantic representation for them with a simple syntactic description.

-Chapter 4 gives a sample representation for numbers, while discussing many incidental issues.

-Chapter 5 discusses the problem of how to detect conceptual phrases.

-Chapter 6 summarizes the results of this effort.

2. Observations about the Behavior of Conceptual Phrases

While investigating the nature of conceptual phrases, it seems desirable to examine the existing Parsifal rules for lower-level constituents, and attempt to find properties and constraints that might allow better representation. The existing grammar includes rules for parsing numbers, times, dates, and proper names. In addition, rules have been constructed for parsing addresses, although they are not in the current grammar. These constructs are all 'lower-level' in the same sense as discussed previously, since they are all constituents of noun phrases.

Inspection of the grammar reveals that the rules for all these constructs follow the same general pattern. A packet of rules for each major type of constituent is introduced by an attention-shifting rule. The rules in this packet attach various constituents from the buffer to a node on to the stack. When no other rule applies, there is one default rule that deactivates the packet and restores the buffer.

It is unusual that each set of rules is introduced by an attention-shifting rule, considering that there are only two other attention-shifting rules in the entire system (NGSTART, which detects and parses noun groups, for use by clause-level rules, and POSSESSIVE-DET, which makes possessives 'look like' determiners to the noun-group parsing rules). As is noted in Appendix I, attention-shifting rules exist to allow the parsing of a phrase to be 'transparent' to the other rules in the system, which refer to the resulting constituent as if it were inserted into the buffer by the lexicon.

A noticeable difference between the rules for lower-level constituents and the other rules in the grammar is that many of the operations they perform consist

of putting objects in registers and performing operations on the contents of registers. The only purpose of these actions is to semantically interpret the phrases parsed by the rule; they do not affect the structure of the final parse tree, nor do they influence the sequence of rules applied during remainder of the parse. Unlike the clause rules, there are no case rules for interpreting nodes built by the lower-level rules, since all semantic interpretation is performed by this manipulation of registers.

Yet another way these rules differ from most other rules is that many features used by the lower-level rules are semantic in nature. The rules for the highest level constituents (i.e. clause-level rules) test features that are clearly syntactic categories, or for the presence of a specific word. The rules for noun phrases also deal with features that are either syntactic categories or tests for individual words; however, there are some features (such as "quant", for a noun quantifier) that seem to reflect semantic properties of the words in addition to purely syntactic characteristics.

The rules for numbers, however, manipulate features such as: "ones" for a natural number less than ten, "tens" for words such as "twenty" "sixty", etc., "hundred+" for a number with the word "hundred" in it, and so on. Other than the tests for specific words, there are very few features in the number-parsing packet that are purely syntactic in the same sense as the features for higher-level constituents. This suggests that the syntax and semantics of lower-level constructs are more closely related than for, say, clause-level constructs. Specifically, the parsing process for lower-level constituents is primarily directed by semantic, not syntactic, categorization.

In addition to these observations, there are other facets of the behavior of lower-level constituents that might help us understand their true nature. These constituents behave like single lexical items in almost all contexts. For example, proper nouns behave exactly like nouns in all cases (justifying their designation), even though they may consist of several different words, or even several different constituents, each with its own substructure (for example, "Percy Flumbatz Memorial High School" could be analyzed as having the structure [[Percy Flumbatz] [Memorial [High School]]]). Also, numerals are treated exactly like single-word quantifiers in almost all contexts (except when they are used as arbitrary identifiers), even though they may contain more than one word.

Another aspect in which the behavior of low-level constructs differs from the behavior of higher-level constituents is that they do not appear to be a 'closed class'. Clause and noun phrase level syntactic structures are part of the permanent structure of the language. It is unlikely that a new top-level sentence structure would be added to a language under ordinary circumstances, or that a new order for noun-group constituents would become part of a language. It seems to be very difficult for a native speaker of a language to acquire the use and understanding of a new high-level construct in any natural way. On the other hand, it is not at all difficult to learn a different way of communicating a date (e.g. "23 June 79" instead of "June 23, 1979"), or to learn a new kind of name that provides different types of information.

The important point to abstract from these observations is that conceptual phrases represent a cross between clause level structures (such as complete sentences) and individual lexical items. They have the clause-like characteristics of

structure and syntax, but also have properties that are 'lexical' in nature, such as representing single semantic classes, being freely added to the language, and filling the syntactic role of single words. The fact that Parsifal fails to recognize this fundamental difference accounts for the lack of perspicuity in the current grammar for numbers, names, times, and other conceptual phrases.

It should also be noted that the meanings of conceptual phrases could be conveyed using the general syntactic mechanism for clauses and noun phrases. Indeed, they may well have developed from such an approach. Consider the derivation of the name John Smith ("John the smith") or Fred Johnson ("Fred, son of John"), which have been derived historically from shorter constituents combined by higher-level syntactic mechanisms. Consider another type of conceptual phrase: numbers. In Britain, it is not unusual to hear phrases like "several millions of pounds" or even "five millions of dollars", where the "millions" is regarded more as a noun than can be quantified in the same manner as any other noun, including the application of rules of number agreement. This, in conjunction with the vestigial presence of the optional word "and" in number phrases, makes it clear that numbers have been thought of historicaly as smaller constituents conjoined by the mechanisms of clause and noun-phrase-level grammars.

If conceptual phrases were communicated using the same syntax as higher-level phrases, the only difference between them and other syntactic structures of the same form would be that they correspond to objects of a single semantic class. This leads to the view that the unique syntax that seems to exist for each conceptual phrase is an 'optimized' mechanism for communicating them. The additional constraints on the semantics of conceptual phrases allow some

syntactic information to be left out that would otherwise have been provided by the 'standard' grammar; yet the ability of the listener to recover all of the useful semantic information is retained.

Of course, the constraint that a phrase must represent a single semantic class is not necessarily sufficient to allow a more efficient communication protocol. In addition, the members of the semantic category must have enough in common structurally so that significant assumptions can be made about the role of various constituents being communicated. For example, if addresses in the United States were not formalized at all, and consisted simply of descriptions of locations ("the second white house along the small alley next to highway that runs by the Flumbatz place"), it would not be possible to communicate them efficiently.

Now that we have arrived at some conclusions about the general nature of conceptual phrases, the original engineering problem of how to represent them in a manner that exploits their apparent simplicity still remains unsolved. The assertion that each type of conceptual phrase has a corresponding optimized syntax does not provide a method for deriving that syntax. It also does not provide a method for 'decoding' conceptual phrases. The next chapter is devoted to developing techniques that do.

3. A Combined Syntactic and Semantic Representation

3.1 Introduction

This chapter presents a technique for specifying the syntax of conceptual phrases in terms of their semantic representations. A prerequisite for this feat is the ability to represent the semantics of conceptual phrases, preferably by a means that has been implemented on a computer. There have been numerous attempts to design 'knowledge representation languages', with varying degrees of success. Some of the better known systems (in the Artificial Intelligence community, at least) are KRL [1], FRL [8], OWL [6], and NETL [3].

I selected the language FRL ('Frame Representation Language') for the purposes of this thesis, because it is implemented in MACLISP, which makes it possible to integrate it with Parsifal, and it is also (relatively) computationally efficient. Although FRL has been criticized on the grounds of insufficient expressive power [2] [9], it includes most of the features common to knowledge representation languages, and I have found that an FRL subset is sufficiently expressive for representing conceptual phrases. Most of the remaining discussion of semantic representation will be in the vernacular of FRL, so the reader unversed in it is urged to read the description in Appendix II before continuing.

3.2 Protocols for Frame Communication

In the previous chapter, it was determined that the major distinction between conceptual phrases and higher-level constituents is that everything that can be expressed by a particular type of conceptual phrase belongs to the same semantic class. In FRL terminology, every frame expressed by a single type of conceptual

phrase is an *instantiation* of a generic *frame*. The approach used to combine the syntactic and semantic descriptions of conceptual phrases is to augment this generic frame with a syntactic description, which can be thought of as a formula for encoding and decoding the information in the frame.

I will defer the details of the syntactic description for now, in order to develop a detailed model for the process of communicating a particular instantiation of a generic frame. This model will clarify the reasoning behind the form of the syntactic description.

Consider the following situation: two people are attempting to communicate via speech. The 'speaker' wants to communicate a particular frame to the 'listener' (i.e., the speaker has a specific frame in mind, and wants the listener to 'build' that frame). For the time being, assume that listener already knows that this frame is an instantiation of a generic frame, and what the generic frame is. Both communicants have agreed on a protocol for communicating any instantiation of this generic frame (because they speak the same language). What type of protocol would suit the purpose of communicating the frame?

One simple, though inefficient, protocol would be to transmit each slot name of the frame immediately before or after its value. This would ensure that the listener could construct a duplicate of the speaker's frame without difficulty. However, this protocol fails to exploit the information that could be passed, implicitly, by the order in which the slot values are communicated.

An improved protocol would specify the order in which slot values should be communicated. The amount of information transmitted would be halved, since the slot names would no longer be transmitted. However, this protocol fails to take

advantage of the knowledge that some slots are much more likely than others to have a particular value and that the communicants may not know or care about the values of other slots. In either case, the value need not be communicated explicitly. In the first case, the absence of a transmitted value would indicate to the listener that a default value should be used, and in the second case, communicating the value would be pointless. This would amount to a major loss in efficiency if most of the slots didn't need to be communicated most of the time.

A protocol is needed that can take advantage of the above optimizations in order to communicate with maximum efficiency. I propose the following protocol:

1. The values of slots should be communicated in a fixed order.

2. Reasonable defaults should be assumed for slots where one particular value is clearly more probable than any others.

3. The optional communication of slots may make it difficult to figure out to which slot a certain value corresponds. If there is any ambiguity, another word should be transmitted either immediately before or after that value to indicate the proper slot (this extra word will be referred to as a *synchronizer* later).

3.3 The Syntactic Description

This thesis takes the position that the above protocol is more than just an engineering exercise. In fact, the syntactic specification will be based on the assumption that this is actually how conceptual phrases are communicated by people.

The specific syntactic specification is tailored to the needs of the listener, who must decode the conceptual phrase. The listener must be able to compare constituents in his 'buffer' (under the assumption that he has a mental Parsifal in operation) with the restrictions specified in the frame on the possible values of

each slot. Consequently, the syntactic specification for the frame must provide for tests on the contents of the buffer to check whether a given constituent can 'fit' into a particular slot in the frame. Also, he must be able to use synchronizers to select the correct slot for a constituent.

The specific representation I have chosen meets all of the above requirements. The frames are ordinary FRL frames, but there is an additional specification for the syntax of any frame (there is nothing to prevent two frames from sharing a syntactic specification, or one frame having multiple syntactic specifications). The syntactic specification consists of an ordered list of 'slot specifications' (in the same order as the slots are to be communicated). Each slot specification consists of a buffer pattern associated with name of one of the slots in the frame. The buffer patterns resemble the buffer patterns in normal Parsifal grammar rules except that they never examine a node on the stack.

If the 'listener' in the earlier communication scenario is Parsifal, with this specification technique for conceptual phrases, the communication process would proceed according to the following simple algorithm:

1. create a generic frame for this conceptual phrase and put it on the active node stack (assume that frames can be nodes).

2. compare each slot specification (pattern/slot pair) of the frame's syntax with the buffer. If the pattern matches, put the contents of the first buffer cell in the specified slot, and throw away all other constituents matched by the pattern. If the pattern does not match, do nothing and continue to the next pattern.

3. when the entire pattern has been exhausted, pop the frame currently on the stack into the first cell of the virtual buffer.

The stack should be in exactly the same state immediately after the execution of this algorithm as immediately before. The effect of the rule is to

collect all the words of the conceptual phrase into a single frame, which replaces them in the buffer. From the point of view of the higher-level rules, the frame might as well have come directly from the lexicon.

3.4 Relation between the Protocol and the Syntactic Description

It is important to note that this algorithm does not allow the full range of syntax permitted by the general protocol presented earlier. For example, the restriction that only the contents of the first buffer cell can be put in a slot (see step 2) is a simplification based entirely on the observation that it seems to work empirically in English.

There is another constraint on the algorithm that does not exist for the protocol, but this one has independent justification. Remember that the protocol requires synchronizers when the correct slot for a value is not obvious from the conditions specified on that slot's value (via the $REQUIRED facet). The algorithm does not look through the possible slots, testing arbitrary predicates on a buffer constituent. Instead, the conditions must be checked entirely by feature tests on the contents of the cells of the buffer.

Theoretically, the features in the buffer could be made arbitrarily complicated, to reflect the results of computing all the predicates that occur in all the slots of all the frames. This would allow the feature test to have the same power as the process of testing all the predicates during the matching process (in actuality, this is just a pre-computation of the predicates). This is counterintuitive, computationally inefficient, and inconvenient, since it would require that all the features on constituents in the lexicon be updated whenever a new condition is placed on a slot. The process of testing the actual predicate at 'parse-time' instead

of 'definition-time' (by analogy with compilers) would probably be better.

However, there is a computational justification for testing features that do not represent all the potential complexity of the predicates on a frame's slot. The $REQUIRED predicates can be arbitrarily complicated. For a human to parse a conceptual phrase in real-time, the complexity of the tests must be bounded in some way. If the tests can only check for an appropriately constrained set of features on buffer constituents, this requirement is fulfilled nicely. Even if the semantic restrictions on the value of a slot are not exactly equivalent to some simple feature test, it is possible that these restrictions can approximately be met by such a test.

This aspect of the algorithm is consistent with the observation of the previous chapter that the semantic objects corresponding to a conceptual phrase must be constrained in some way besides simply fitting into a single semantic class. That constraint can now be expressed in terms of the computational effort required to determine whether a particular value can 'fit' into a particular slot. As observed in chapter 2, the features tested by the current rules for conceptual phrases correspond either to tests for particular words or for semantic features (i.e. semantic categories). Perhaps the additional constraint of that chapter is that any frame expressible by a conceptual phrase must consist of a finite set of predesignated slots, whose conditions may be approximated by tests for individual words or the semantic classes of constituents.

3.5 The Representation and Meaning

This approach to the representation of conceptual phrases has a significant implication with respect to the nature of meaning. A commonly held view (even when not stated explicitly) is that a single 'concept' has a single canonical semantic representation, which is used whenever the meaning of something is needed (for communication, formation of other concepts, or some form of manipulation). This suggests a picture of natural language understanding as a transformation from a string of words in the language to this idealized concept.

The representation here would run into severe problems when evaluated in such a framework. Since the semantic representations must be chosen to reflect the structure of the syntactic representations, they may not be in the idealized form that tries to be all things to all people. If the structure of the semantic representation is not chosen to reflect the syntactic structure, the rules for translating from the semantics to the syntax, and vice versa, become extremely complicated.

In order to be able to use the semantic representation as a basis for the syntax of a conceptual phrase, and still retain the suitability of the semantics for other types of manipulation (e.g. structure building, computation, or deductive inference), we will have to take a different view. One view that seems to be particularly appealing is that the 'meaning' of something is not just a single semantic structure: it is a set of such structures, each suited for a special purpose or group of purposes. Such a system would need rules and procedures for translating among the various special-purpose representations.

The frames we have been referring to as 'semantic representations' are

actually just a part of such a system. This is not to say that the only thing they are good for is communication (this would render the term 'semantic' meaningless). First, the existence of the generic frame provides a point of reference for finding various associations with an object. For example, once an address phrase has been parsed into an address frame, we have immediate access to all the generic properties of addresses. We also have a frame to match against memory to see if one can derive any additional information from knowledge about this particular address. Finally, such frames (or something vary close) could conceivably be the basic item manipulated during verbal reasoning, when people translate their thoughts into language to help them 'think things through'.

4. Numbers: A Sample Representation

4.1 A Representation for Numbers

This chapter develops a representation for English numerals (American dialect), in the form of syntactically augmented frames as described in the previous chapter. Although most of the discussion in this chapter relates specifically to numbers, it should understood that many of these issues are common to the representation of many other types of conceptual phrases. The representation of numerals was chosen as an example because they are a particularly clear and well understood form of conceptual phrase.

Below is a sample set of frames for natural numbers (less than one trillion). In this representation, numerals are actually represented as several different types of conceptual phrases, some of which can be contained within others. The digits "one" through "nine" are all in the lexicon as members of the semantic class of ONES. Numbers between twenty and ninety, inclusive, that are multiples of ten are in the semantic class of TENS. The semantic class SMALL-NUMBER also contains all legitimate combinations of TENS and ONES, such as "thirty six" and "ninety nine". Also, there are some SMALL-NUMBERs entered in the lexicon, namely, the numerals "ten" through "nineteen". The conceptual phrases for TENS and ONES can be constitutuents of a SMALL-NUMBER. Another conceptual phrase is HUNDREDS: any SMALL-NUMBER followed by the word "hundred", optionally followed by another SMALL-NUMBER. Note that this allows phrases such as "fifty four hundred", which are not formally correct, but are almost universally used. Finally, the BIG-NUMBER frame includes those phrases that have the words "thousand", "million", or "billion" in them. This system cannot recognize numbers greater than "999 billion".

The frame for a SMALL-NUMBER is shown below. All LISP functions have been paraphrased in English, and is delimited by '%'.

```
(FRAME SMALL-NUMBER
   (TENS ($DEFAULT (ZERO))
         ($REQUIRED (%the value must be a kind of TENS%)))
   (ONES ($DEFAULT ZERO)
         ($REQUIRED (%the value must be a kind of ONES%)))
   (AKO ($VALUE (NUMBER)))
   (QUANT ($IF-NEEDED (%calculate the value by adding the
                       values of the TENS and ONES slots%))))

(SYNTAX SMALL-NUMBER
   ((=TENS) TENS) ((=ONES) ONES))
```

The $REQUIRED slots in the above frame are entirely redundant, since they correspond exactly to the features specified in the syntactic description of the frame. The only reason they are defined here is to emphasize that, according to the analysis of the last chapter, the feature tests of syntactic description are an efficient approximation to constraints on the possible values of a slot. The constraint is expressed in the $REQUIRED facet, even though the syntactic description enforces it exactly, in this case.

For the conceptual phrase "fifty six", this frame would test if the constituent in the first place of the buffer had the feature TENS, put it in the TENS slot of a SMALL-NUMBER frame, then test "six" for the ONES feature, and put it in the ONES slot. The constituents for "fifty" and "six" would be frames themselves, but these frames are 'prefabricated', and associated with root words in the lexicon.

The $DEFAULT facets in the TENS and ONES slots of the frame indicate that if no value is available to put into a frame explicitly, the hearer should assume that it has the value "ZERO", if the value is needed (actually, in this model,

getting a default value from a frame is a more basic action psychologically than 'assuming' something).

The result of parsing the conceptual phrase "fifty six" is the frame:

```
(FRAME SMALL-NUMBER11
       (TENS ($VALUE (FIFTY)))
       (ONES ($VALUE (SIX)))
       (AKO ($VALUE (SMALL-NUMBER)))).
```

As in any other FRL frame, this frame inherits any properties that are not explicitly defined in the frame from other frames in the AKO hierarchy. In this case, the frame is an instantiation of the SMALL-NUMBER frame, so among the inherited properties would be the default values for its slots and the $IF-NEEDED facet of the QUANT slot.

Notice that the QUANT slot does not appear in this frame. The value of the QUANT slot is the 'computer' concept of a number (i.e. LISP 'fixnum'). There are at least two ways this could be computed. The first would be to have a LISP program associated with both of the TENS and ONES slots which will calculate the value and add it to the contents of the QUANT slot. The other approach, which is the one used here, is to have a LISP program which calculates the value of the QUANT slot only if it is asked for. This method is more compatible with the view of 'meaning' taken in the last chapter, that a concept is a group of different representations, each suited for specific uses. If the the number is going to be used for computation after being parsed, the machine representation is clearly better. On the other hand, if the number is used to find associated knowledge (e.g. a year in history), the computer representation is not needed, and should not be computed.

The frame for HUNDREDS is:

```
(FRAME HUNDREDS
   (NUMBER-OF-HUNDREDS ($REQUIRED (%must be less than 100%)))
   (REST-OF-HUNDREDS ($DEFAULT (ZERO))
                     ($REQUIRED (%must be less than 100%)))
   (AKO ($VALUE (NUMBER)))
   (QUANT ($IF-NEEDED (%100 times the quant of the value of the
                       NUMBER-OF-HUNDREDS slot, plus the quant
                       of the value of the REST-OF-HUNDREDS
                       slot.%))))

(SYNTAX HUNDREDS
   ([* IS ANY OF SMALL-NUMBER, TENS, ONES] [=*HUNDREDS]
    NUMBER-OF-HUNDREDS)
   ([* IS ANY OF SMALL-NUMBER, TENS, ONES]
    REST-OF-HUNDREDS))
```

(In the pattern for each slot, "* IS ANY OF" means that the pattern should match if *any* of the features are present. Also, a pattern of the form '*FOO' means 'match on the presence of the word "foo".)

In this case, the presence of the word "hundred" is a synchronizer; it differentiates between the conceptual phrase for the NUMBER-OF-HUNDREDS slot and the conceptual phrase for the REST-OF-HUNDREDS slot. Also, note that the SMALL-NUMBER rule should be transparent (e.g. "fifty six" should appear to be a single lexical) to this rule, since the HUNDREDS rule can have a SMALL-NUMBER as a constituent.

The generic frame for a BIG-NUMBER is represented as follows:

```
(FRAME BIG-NUMBER
   (BILLIONS ($DEFAULT (ZERO))
             ($REQUIRED %the value must be less than 1000%))
   (MILLIONS ($DEFAULT (ZERO)))
   (THOUSANDS ($DEFAULT (ZERO)))
   (REST-OF-BIG-NUMBER ($DEFAULT (ZERO)))
   (AKO ($VALUE (NUMBER)))
   (QUANT ($IF-NEEDED
            (%take the sum of the slots after multiplying the
             QUANT of the value of each one by the appropriate
             power of 10%))))

(Note: the $REQUIRED facets of the other slots have been
left out to avoid needless repetition)

(SYNTAX BIG-NUMBER
   ([* IS ANY OF HUNDREDS, SMALL-NUMBER, TENS, ONES] [=*BILLIONS]
    BILLIONS)
   ([* IS ANY OF HUNDREDS, SMALL-NUMBER, TENS, ONES] [=*MILLIONS]
    MILLIONS)
   ([* IS ANY OF HUNDREDS, SMALL-NUMBER, TENS, ONES] [=*THOUSANDS]
    THOUSANDS)
   ([* IS ANY OF HUNDREDS, SMALL-NUMBER, TENS, ONES]
    REST-OF-BIG-NUMBER))
```

This frame has many more synchronizers than the other frames in the numeral system, since there are many adjacent slots with identical semantic conditions. Also, this rule has a case of a syntactic condition approximating a semantic condition on a slot. Phrases such as "fifteen hundred" are allowed as constituents, (allowing "four million fifteen hundred thousand") since they are legitimate HUNDREDS. This situation could have been avoided by requiring a HUNDREDS to have a ONES as its first component, instead of a SMALL-NUMBER, or by creating an entirely new conceptual phrase for parsing numbers of the form ONES followed by a HUNDREDS, etc. I chose the representation presented here since, on the basis of informal experimentation, it seems that people did not notice the incongruity of numbers such as "one million fifty six hundred thousand five" until asked to write them

down or perform arithmetic on them[1].

Finally, I should note that this representation has left out at least two minor details of the grammar for natural numbers. First, conceptual phrases in the HUNDREDS category have an optional "and" after the word "hundred" (e.g. "six hundred and fifty seven"). This can be expressed very easily in the representation by allowing a slot-specification with no slot name, which would effectively discard the "and".

Alternatively, an *additional* syntactic description could be constructed for the HUNDREDS, having a slot specification of the form

```
(SYNTAX HUNDREDS
        ((=SMALL-NUMBER, TENS, ONES) (=*HUNDRED) (=*AND)
         NUMBER-OF-HUNDREDS)
        ((=SMALL-NUMBER) REST-OF-HUNDREDS))
```

accomplishing the same goal.

Another irregularity that is not accounted for here is the occurrence of phrases such as "a million and ten thousand", where a leading "one" can optionally be an "a". This should probably be taken care of elsewhere in Parsifal; since "a" and "one" have the same quantificational properties, perhaps the lexical phase of the parser should convert them into the same constituent.

4.2 The Validity of the Representation

Due to a lack of evaluation procedures, it would be very difficult to determine whether the numeral representation presented is 'the correct one'. A supporting point for this type of representation is the perspicuity of the representations of individual conceptual phrases. The representation given here

---

[1] A. Van Katwijk [4] makes a similar observation to justify the omission of context-sensitive rules in a grammar for Dutch numerals

manages to express a broad class of numerals fairly succinctly. Even if another system could arrive at a more perspicuous representation for numerals in one of the domains of syntax or semantics, it seems unlikely that this could be achieved without greatly complicating the other domain. In addition, considering the earlier observation that conceptual phrases are an optimized means of communicating a member of a constrained semantic class, the fact that this representation manages to capture the inter-relationship between the semantic and syntactic knowledge for these expressions increases its attractiveness.

Additionally, a numeral in this representation can be useful for operations other than communicating or retrieving associated information. Although it would not be as suitable for some arithmetic operations as another representation might be, it would be adequate for many common operations. For example, two SMALL-NUMBERS can be compared easily in size by the simple algorithm of comparing the TENS (via 'table lookup') and if they are the same, comparing the ONES. It is also very easy to compare cardinal numbers of different semantic types. In general, all the members of any semantic class are either greater than or less than all the members of any distinct semantic class (the exceptions are comparing TENS and SMALL-NUMBERS, which can be tested by comparing the TENS slot of small-numbers, and "fifty six hundred" vs. "five thousand six hundred".[2]).

Also, simple addition and subtraction of numbers in this representation is fairly simple (both in terms of computation and short-term memory), as long as no 'carrying' is required. Indeed, the fact that these operations are so much more

---

[2] If it could be shown that people empirically have greater difficulty in comparing numbers like "forty five hundred" with "six thousand three hundred" than with "sixty three hundred", it would lend support to the hypothesis that the semantic representations for these forms are different

difficult when 'carrying' is required, may indicate that this representation is the 'default' for general-purpose conceptualization.

5. When to Expect a Conceptual Phrase

When the protocol for frame communication was developed in Chapter 2, it was assumed that the listener knew which generic frame was being communicated by the speaker. Until now, the method by which this frame is recognized has remained unspecified. Since the syntactic description of the frame does not contain enough information, a priori, to determine when the frame applies, we must provide either some external indication of the 'triggering' condition, or some heuristic for deriving that condition from the syntactic description.

As an added complication, the recognition process cannot be entirely syntactic in nature. Consider the phrase "nine thirty" in isolation. This could be a time of day (9:30), a price ($9.30 or $930), part of an address ("930 Percy St."), or almost any type of number, depending on the context and the domain-specific expectations of the listener. This is exactly the same problem that occurs when trying to determine which word sense to associate with a given word, and once more indicates the partly lexical nature of conceptual phrases. Assuming that discourse context is not sufficient to guarantee that only one possible conceptual phrase can occur, there must be some syntactic preference involved.

For engineering purposes, it would probably be adequate to construct the triggering information for these rules by hand and represent it explicitly as a Pidgin rule. It would be much more convenient, however, to have an automatic 'trigger generator' produce the triggering information from the specification of the frame. In either case, some decisions must be made as to how the rules should interact, since this interaction has a major effect on the interpretation of the rules.

As an example of how the specific triggering conditions for a frame can

affect the final result of the parse, consider the specification for numerals presented in the last chapter. If we adopted the philosophy that a frame should trigger whenever something appears that fits in one of its slots, the following scenario could occur: the word "one" could occur at some point in the input. It would be associated with the ONE frame in the lexicon, which would be in the semantic class ONES, and insert it into the buffer. The SMALL-NUMBER frame could then trigger, since the object in the semantic class ONES can fill the ONES slot of that frame. This would leave a SMALL-NUMBER in the buffer, associated with the phrase "one". Similarly, the HUNDREDS and BIG-NUMBER frames would trigger, and the result of the process would be the appearance of a BIG-NUMBER constituent in the buffer, in place of the phrase "one".

This type of behavior has problems in the interaction with the rest of the grammar. Since cardinal numbers are frequently used as quantifiers, the rule for number-agreement applies. Any noun quantified by the number "one" is singular, while a noun quantified by any other number is plural. If all cardinal numbers were BIG-NUMBERs distiguished only by the contents of their frames, the number-agreement rule would have to examine the *contents* of the BIG-NUMBE ' frame in order to function correctly. However, number agreement seems to be more of a syntactic than semantic phenomenon[3] . In addition, if semantic description and syntactic descriptions for SMALL-NUMBERs, for example, already exist it seems artificial to say that they have no independent existence, except to trigger the HUNDREDS rule.

A less liberal triggering philosophy that required the phrase for, say,

---

[3] consider the acceptability of "1.0 cavities"

HUNDREDS to contain at least one other constituent besides a single SMALL-NUMBER would have a much different result. If a similar triggering criterion held for the other frames, each different type of number could without being part of another frame.

Even if the representation for numbers could be reformulated to avoid this difficulty, it would still be difficult to control interactions between frames, in general. There is no restriction in the representation to prevent different frames from having slots that permit the same types of constituents, under the same conditions. If this situation occurs in the grammar, and it seems probable that it will, there will have to be some general principle to determine which rule should trigger.

It seems to me that the proper behavior of the rules should be to prevent frames that contain only one constituent from occuring, whenever this is practical. I propose the following more specific principles for determining the triggering conditions for a frame:

1.All frames must trigger on at least two buffer constituents.

2.If two frames can trigger at the same time, then

> If one of the frames is a constituent of the other ('constituent' here means related by the transitive closure of the 'immediate constituent' relation), it should trigger first.
>
> If neither is a constituent, hope that some other mechanism, such as discourse context, will give one rule preference.
>
> If each rule can be a constituent of the other, the representation is probably wrong anyway. I'm certainly not going to worry about it.

If the triggering rules interact so as to adhere to this principle, the effect

would be that the frame that triggers would be the 'lowest-level' conceptual phrase (thinking of the frames in terms of a constituent hierarchy). There should be no need to allow a frame that contains only one constituent, since that constituent would be a single semantic entity, and should be useful by itself.

An implementation problem is raised by this approach. Since we have been assuming that the conceptual phrases are parsed into frames in a manner transparent to the other rules (including the ones for parsing other conceptual phrases), the only really convenient trigger mechanism is attention-shifting rules (which explains the use of attention-shifting rules in the current grammar). Unfortunately, the implementation of attention-shifting rules in Parsifal constrains them to trigger on a single buffer cell, in order to prevent violations of theoretical constraints on 'look-ahead' in the buffer. It would still be possible to implement triggering by having an attention-shifting rule match on the first cell of the pattern for triggering the frame, and then have other (normal) rules check the rest of the pattern against the buffer, and restore it if none match. This addition to the complexity probably indicates a missed generalization, either in Parsifal or in our representation.

Additionally, from an engineering point of view, it is not convenient to have to examine all the other rules in the system whenever a new conceptual frame is learned. At this time, the problem of how best to specify the triggering mechanism remains open.

6. Conclusions

6.1 Where Conceptual Phrases Fit In

One of the major unsolved problems of both linguistics and artificial intelligence is to define the relationship between language and knowledge. The solution, if there is one, would have profound impact on both fields, and probably on many other fields as well.

This research represents an attempt to find a solution to that problem on a very small scale -- to develop a representation that captures the relationship between the syntax and the semantics of what I have termed 'conceptual phrases'. The result is a specification that consists of a frame representation of an object, augmented by a syntactic description of the frame.

Conceptual phrases fill a niche between the various phases of the parsing process. The different types of linguistic constructs, measured along the axis of complexity in semantic interpretation, can be seen as a spectrum. At one end of the spectrum, there are 'sentences', which are exceedingly general constructs in terms of the meanings they can express. The users of the language pay a price for this generality (as with most computational tasks), in terms of verbosity and the difficulty of semantic interpretation.

At the other end of the spectrum are individual words. The process of associating a word with its meaning is very simple. Simply look it up in the lexicon. Words, by themselves, have a very limited range of meaning, however.[4]

---

[4] when I say "word" here, I mean an individual lexical item corresponding to a single word in a sentence, where all the syntactic and discourse constraints have already acted to select the correct word sense from a fixed list of alternatives

Conceptual phrases are somewhere between these two ends of the spectrum. They are more general than words, but less general than clauses, for communicating concepts. Their range of meaning is sharply limited, but there are many more possibilities than for single words. The process of semantic interpretation is much simpler than for sentences, but more complex than for single words. It is interesting to speculate whether there are additional intermediate stages, each with its own special tradeoff between expressive power and efficiency.

There seem to be at least two major schools of thought, among linguists and in artificial intelligence, about the relationship between syntax and semantics. One school contends that the processes of language understanding and acquisition are merely facets of a general cognitive mechanism, and that syntax is an arbitrary set of rules for converting knowledge structures to linear streams of words, for the purpose of communicating it. The other school believes that syntax and semantics are separate (though connected mechanisms), and that people have a special 'cognitive processor' or processors that is devoted to the manipulation of language.

Parsifal has been most strongly influenced by the second line of thought, yet the mechanism proposed herein is highly reminiscent of the first philosophy. The combination of Parsifal's grammar language for higher-level constituents and this specification for the lower-level ones is a synthesis of the two viewpoints.

6.2 directions for future research

In order to have a practical system for specifying conceptual phrases, there are some problems which still must be dealt with. First, there is the problem discussed in Chapter 4 of how to generate 'triggering' information from the augmented frame specification. This is part of a larger question about exactly how

the frames should interact. Another task would be to determine how much power is required in the syntactic description, preferably stated in terms of linguistic universals rather than as ad-hoc observations about conceptual phrases in English.

There are interesting linguistic questions about conceptual phrases, if the analysis given here is correct. The communications model developed in Chapter 2 seems to imply that the process of generating a conceptual phrase is different from the process of composing a sentence. Can the representation given here be modified to make it bi-directional (suitable for the speaker as well as the listener)?

Additionally, there is the question of how conceptual phrases are learned. Since the syntax seems to be simpler than for higher levels, are they easier to learn? Chapter 2 mentions that conceptual phrases can be added to the language relatively freely. Does this imply that the process of learning conceptual phrases is substantially different than the process of learning other syntactic rules? The principles developed in Chapter 5 could be implemented as an algorithm which checks for potential triggering conflicts with other rules. Does this imply that the decision about how the frames should interact is made when the rule is acquired?

I. A Description of Parsifal

This appendix is meant to provide a description of those parts of Parsifal which are necessary for understanding the thesis. Parsifal is an English parser developed by Mitchell Marcus of the MIT Artificial Intelligence Laboratory. Unless otherwise noted, this description is abstracted from his Ph.D. thesis [5]. The reader is referred to that document for a detailed description of Parsifal and its implications for the fields of linguistics and artificial intelligence.

Marcus calls the underlying philosophy of Parsifal "strict determinism" (not to be confused with the concept of determinism in automata theory, which is far too broad to be of use in this context), but Swartout [10] claims that a more accurate view of the parser's operation would be that it adheres to the "principle of least commitment" -- it defers decisions until the latest possible moment, and when it makes a decision, it considers it final.

In addition to determinism, another goal of Parsifal is to tailor the capabilities of its 'abstract machine' (consisting of the data structures and the operations which may be performed upon them) as closely as possible to the task of syntactic parsing of natural language. This is done in the hope of allowing very brief and powerful definitions of natural languages that should provide insight into the psychological mechanisms at work when a person understands language. In addition, this tailoring should simplify the task of constructing grammars for practical natural language understanders.

Parsifal was designed with the theory of generative grammar very much in mind, and maintains the position that syntactic and semantic mechanisms form a nearly-decomposable system. In other words, there is a separate 'syntactic processor'

that interacts in some limited way with the mechanisms that actually derive the meaning of a construct. This contrasts with many approaches to natural language understanding that assume (implicitly or explicitly) that there is no separate mechanism for syntactic analysis.

Marcus calls Parsifal a "grammar interpreter", since it can be divided conceptually into an "abstract machine" which interprets a set of "grammar rules", which are responsible for directing the parsing process. The abstract machine implemented by Parsifal consists two major data structures.

The first is a buffer, which consists of a series of three "cells" which may contain "constituents". A constituent is any grammatical structure, including the final parse tree, various subtrees and individual words. The rules may insert constituents into the buffer or delete constituents from it. The buffer is a 'compacting buffer', meaning that when an object is inserted, the contents of the buffer will 'move aside' to create an open cell for it, and when an object is deleted, the remaining contents will be shifted to 'close the gap'. A rule may only delete an constituent from the buffer in the process of attaching it to another constituent. Rules are allowed to insert constituents arbitrarily.

The buffer just described should be called the 'virtual buffer', because it is actually a three-cell window into a larger buffer, of unspecified length. No rule may examine, insert, or delete the contents of any cell outside of this window. There is an operation called 'offset' that allows this window to be shifted three or fewer cells to the right, and an operation 'restore' which moves the window back to its position before the last 'shift'. This capability is only used by "attention-shifting rules", which will be explained slightly later.

The second data structure used in Parsifal is a stack. It functions as temporary storage for constituents being constructed. The only way anything can be "pushed" onto the stack is by the creation of a "node". When a node is created, it becomes the "current active node", and stays on the top of the stack until another node is created, or until it is removed. Nodes are a type of constituent, and the terms will be used interchangeably in the rest of this thesis.

A rule can attach a constituent in the buffer to the current active node. When this occurs, the constituent is removed from the buffer (this is the only time the delete operation can occur). Nodes may be removed from the stack, in which case they are inserted into the first position of the buffer (relative to the virtual buffer start), and the next node on the stack becomes the current active node. These operations are used in building higher-level constituents from lower-level ones. Typically, a rule will create a node, attach several constituents to it from the buffer, and then take the node off of the stack and put it in the buffer. The effect of this is to combine several constituents in the buffer into a single node, which replaces them in the buffer.

The contents of the buffer and stack are manipulated by grammar rules, written in Pidgin, Parsifal's grammar definition language. Grammar rules are defined in the form of pattern/action rules. A pattern consists of a description of the contents of the buffer and stack. This description consists of a boolean combination of tests for the presence or absence of "features" on the node occupying that position. These features can be manipulated by the "action" part of the rules, but are also associated with words in the lexicon, and may be modified by the morphology of a word before it is inserted in the buffer.

Grammar rules have two other important attributes. Each rule has one or more "packets" associated with it. For a rule to take effect, not only must its pattern match against the buffer, but its packet must also be "active". Those packets that are active at any time are associated with the current active node, and when a new node is pushed, the packets associated with the old node become inactive until it becomes the current active node again. Packets may also be activated and deactivated explicitly by the action parts of rules. The other attribute is the rule's "priority". If two rules match, and both are in currently active packets, the one with the lowest priority is run.

Finally, there is a special type of rule called an "attention-shifting" rule, which operates in a manner somewhat different from "normal" rules discussed above. An attention-shifting rule has a buffer specification which examines only a single cell. If the any constituent in one of the virtual buffer cells matches the pattern, the rule has the effect of shifting the virtual buffer so that buffer cell will be the first in the window. The virtual buffer will remain shifted into this position until it is restored by some other rule. The parser checks to see if any attention shifting rule matches before it tests the normal rules.

Attention-shifting rules are used to make constituents of one type 'transparent' to rules of another type. For example, rules that operate on "clause level" constituents in the parser have patterns which match on the presence of a noun phrase in a specific cell of the buffer. Since noun phrases usually consist of a combination of lower-level constituents, some set of rules must build them. Ideally, the process of parsing the noun phrase would be invisible to the clause-level rules, so that they could treat an them exactly like individual words.

This effect is achieved by an attention-shifting rule, which detects the presence of a "noun group start", shifts the buffer, activates a packet of rules which parse the noun group. A final rule restores the buffer.

Currently, there is a rudimentary case frame system which operates in parallel with the parsing process. The details of this system are irrelevant to the purposes of this paper, beyond the fact that they are the only form of semantic interpretation existing in the basic Parsifal system.

The result of the parsing process is what Marcus calls an "annotated surface structure", which consists of a tree of nodes, representing various types of phrases. Each node is labelled with a set of features that describe the phrase it dominates (for example, a single noun might have the feature "ns" on it, meaning that it is a singular noun. This feature is used by the rule which checks for number-agreement between modifiers and nouns). The process of parsing proceeds something like this:

> The word is analyzed by the morphology program, which attempts to find its root word in the lexicon. The root has various features associated with it. The morphology program may modify some of these features or add some of its own, depending on the word ending (for example, the morphology program may decide that because a word ends in "-s" and is a noun, that it is plural, in which case it will mark it with the "npl" feature). This root word, with features, is inserted into the left-most empty cell in the virtual buffer.
>
> Some rule may trigger off of the buffer at this time. Say, for example, that the first constituent in the buffer is an "the". The the attention-shifting rule "NGSTART" will trigger, activating a packet of rules which attempt to parse a noun phrase. The rule "NGSTART" will create an "NP" node (which is defined to make it the current active node). Successive rules will then attach the contents of various cells in the buffer to the new node, until the end of the noun group is detected. Then the (completed) NP node will be dropped into the buffer, along with all the features placed on it during its construction, and the virtual buffer will be restored to the same absolute position that it occupied before NGSTART triggered. If the sentence is grammatically correct, other

rules will trigger on the completed NP, forming higher-level constructs, until the entire sentence is parsed.

Here are some example rules in the grammar:

```
!RULE IMPERATIVE IN SS-START
[=tnsless] -->
Label c imper, major.
Insert the word 'you' into the buffer.
Deactivate ss-start. Activate parse-subj.!
```

The rule is named 'imperative'. It is in the packet SS-START. It triggers if the constituent in the first cell of the buffer has the 'tnsless' (tense-less) feature (this is the pattern). If the pattern matches, it puts the feature 'imper' and 'major' on the current active node (represented by 'c' in Pidgin rules). It inserts the word 'you' in the buffer (transforming the sentence to a declarative). I then deactivates the packet that it is in and activates the packet for parsing declarative sentences. It is a testimony to the design of Parsifal's abstract machine that this one rule is the only addition to the grammar for declarative sentences that is necessary for it to parse imperatives.

A rule in the grammar for numbers is:

```
!RULE NINETY-NINE IN BUILD-NUMBER
[=tens] [=ones] -->
Label a new num node 99s.
Attach 1st to c as num1.
Attach 2nd to c as num2.
Set the quant of c to
          plus(the quant register of 1st,
               the quant register of 2nd).
Transfer ord from 2nd to c.
Drop c.!
```

This rule matches on the presence of a constituent in the first place of the buffer with a 'tens' feature, followed immediately by a constituent with a 'ones' feature. It creates a new node (putting it on the stack, of course) called 'num', and puts the feature 99s on it. Then it attaches the two constituents from the buffer

to it (deleting them from the buffer). It then calculates the computer word for the number, and drops the completed number into the first place in the virtual buffer.

II. A Description of FRL

FRL (Frame Representation Language) is a general knowledge representation language, implemented at the M.I.T. Artificial Intelligence Laboratory. The following approximate description is derived from "The FRL Manual" by R. Bruce Roberts and Ira Goldstein [8]. Since the FRL system is much more flexible and powerful than this brief description would suggest, the reader interested in its full capabilities is referred to that document.

FRL attempts to provide a practical implementation of Minsky's concept of a 'frame' [7]. The system centers around a data structure called a 'frame'. A frame has a name and 'slots', which can be thought of as named attributes that can hold information about the frame.

Each slot has a number of 'facets' associated with it. Each facet can be associated with a LISP object (often another frame). The facets are used for associating different types of properties with the slots of a frame. There are a fixed number of possible facets, and the only ones relevant here are the "$VALUE", "$DEFAULT", "$REQUIRED", "$IF-ADDED" and "$IF-NEEDED" facets.

The "$VALUE" facet may be associated with any LISP object, which will then be interpreted as the "value" of the slot. For example, if we had a PERSON frame with a NAME slot, we could make the name PERCY into the value of that slot by associating it with the $VALUE facet of that slot.

If an attempt is made to get the value of a frame's slot, and there is no $VALUE facet for that slot, the slot is inspected for a $DEFAULT facet. If there is such a facet, the object associate with it is returned as the value. If there is no $DEFAULT slot, the system attempts to find a $IF-NEEDED facet to the slot. If there

is one, and it has a LISP function associated with it, that function is run. This feature is frequently used to compute values dynamically from other information in the system.

The purpose of the "$REQUIRED" facet is to allow restrictions to be placed on the possible values of a slot. An arbitrary LISP predicate can be associated with the "$REQUIRED" facet of a slot to test any value that is being associated with the "$VALUE" facet of the frame. If the predicate does not evaluate to T when a value is being added, the system complains.

There is one special slot in each frame, called the "AKO" slot ('a kind of'), that stores hierarchical information about the frames. The entire set of frames can be linked together in a hierarchy via this slot. A frame that has another frame as the value of its AKO slot is an 'instantiation' of the second frame.

The AKO hierarchy is used to allow information about a frame to be inherited. In general, a frame will inherit all the properties that are not specified in that frame, from all the other frames above it in the hierarchy. As a simple example of a case in which this feature would be useful, imagine a PERCY frame that is an instantiation of an AMERICAN frame. The value of the CITIZENSHIP slot of the AMERICAN frame could be set to USA, to avoid having to respecify it identically in all the instantiations of the AMERICAN frame. Thus, the value of the CITIZENSHIP slot of the PERCY frame would be USA, without explicitly defining it in the PERCY frame.

References

[1] Daniel G. Bobrow and Terry Winograd.
An Overview of KRL, a Knowledge Representation Language.
Cognitive Science 1(1):3-46, 1977.

[2] Jon Doyle.
Hierarchy in Knowledge Representation.
Technical Report Working Paper 159, MIT Artificial Intelligence Laboratory, 1977.

[3] Scott Fahlman.
A System for Representing and Using Real World Knowledge.
PhD thesis, Massachusetts Institute of Technology, September, 1977.

[4] A. Van Katijk.
A Functional Grammar of Dutch Number Names,
In H. Brandt Corstius, editor, *Grammars for Number Names*, chapter 1.
D. Reidel Publishing Company, Dordrecht-Holland, 1968.

[5] Mitchell Marcus.
A Theory of Syntactic Recognition for Natural Language.
PhD thesis, Massachusetts Institute of Technology, February, 1978.

[6] William A. Martin.
Descriptions and the Specialization of Concepts.
Technical Report TM 101, MIT Laboratory for Computer Science, 1978.

[7] Marvin Minsky.
A Framework for Representing Knowledge,
In Patrick Henry Winston, editor, *The Psychology of Computer Vision*,
chapter 6. McGraw-Hill, 1975.

[8] Bruce R. Roberts and Ira Goldstein.
The FRL Manual.
Technical Report AIM-409, MIT Artificial Intelligence Laboratory, 1977.

[9] Andrew Schulert.
On the Evaluation of Representation Systems for Semantic Knowledge.
Bachelor's thesis, MIT , August, 1978.

[10] William R. Swartout.
A Comparison of PARSIFAL with Augmented Transition Network Grammars.
Technical Report AIM 462, MIT Artificial Intelligence Laboratory, 1978.

[11] Terry Winograd.

Chapter 4

On the Evaluation of Representation Systems For Semantic Knowledge

Andrew Schulert

Schulert argues in this next chapter that a text comprehending system should be able to use the structure of a network of frames to determine the meaning of strings of "noun-noun" modifiers like "water meter cover" and "reading comprehension mastery tests". He argues that much of the information needed to understand to what such noun-noun strings refer seems to be optimally encoded with a frame system.

By attempting to implement such a system within the confines of FRL, however, we have discovered that for such information to be accessible, it must be in some language that can be used for inference by the underlying representation system. Because restrictions on the slots of a frame are arbitrary LISP programs which cannot be easily examined to see what they will or will not accept, FRL seems to be inadequate in this regard.

Table of Contents

I. Introduction

II. Foundations

The Problem of Semantic Representation

Evaluating Representation Systems

III. Syntax

IV. Knowledge Representation

Knowledge Representation Systems

Frame Theory

FRL

V. The Problem, and a Solution

Another Look at the Problem

The Problem in Terms of Frame Theory

The Problem in Terms of FRL

VI. Implications of the Solution (or Lack Thereof)

Implications for FRL

Implications for Frame Theory

VIII. Summary

I. Introduction

This thesis is concerned with the evaluation of proposed representation systems for semantic knowledge. In particular, it is concerned with the representation system proposed by frame theory (Minsky, 1975) and an implementation of frame theory in LISP, Frame Representation Language (FRL) (Roberts and Goldstein, 1977a). It begins by claiming that there is an a priori correct method of representing semantic knowledge, the method by which people represent knowledge. It then claims that, given this model, prospective representation systems can be evaluated by examining the extent to which they mimic the model in their ability to represent solutions to problems which are intrinsic to representation systems (in that they are present regardless of the domain under consideration).

This approach is demonstrated using the problem of parsing noun groups consisting of three or more nouns. A syntactic parse cannot be complete without determining the modifying relationships among the component nouns. For example, how does one determine that the following are the generally accepted context free parses of the indicated noun phrases? (Bracketing two elements indicates that the left element is modifying the right element.)

[[water meter] cover]

[flywheel [adjustment screw]]

It is argued that, regardless of the syntactic constraints, it is necessary to appeal to semantic knowledge for a complete solution. A solution to the problem is sketched in terms of frame theory and FRL, and the implications of this solution are discussed.

The thesis is divided into five sections. The first section explains this paper's approach to evaluating knowledge representation systems. The second section examines the total scope of the problem of parsing noun groups and delimits the semantic problem with which the rest of this paper will be concerned. The third section is a general overview of semantics and knowledge representation systems, describing the type of system in which a solution could be implemented. This section includes a description of frame theory and FRL. With this as a basis, the fourth section attempts a first order solution to the problem using the knowledge structuring primitives of frame theory and FRL. This solution is not complete, but it does contain elements which must be present in any solution. The implication of these elements for FRL, frame theory, and knowledge representation in general are discussed in the final section.

II. Foundations

The Problem of Semantic Representation

This thesis is based on two premises. The first is that construction of a system for the representation of semantic knowledge is fundamentally different from the construction of a system for the representation of knowledge about accounts payable, say, or even symbolic integration or chess. This is because there is, in some sense, a "right" way to represent semantic knowledge. This right way is the way in which people represent semantic knowledge.

One reason that one might give for this being the right way to approach the problem is that people have the only available example of a working

system. This, however, is not valid. If researchers into symbolic integration had taken this approach it is safe to assume they would never have attained their current success. The reason is not simply that people represent semantic knowledge, but rather that the problem is defined in terms of the representation system that people have. The problem of semantic analysis is not the divining of an absolute meaning in the classical sense, but rather the determining of the meaning that would be assigned to an utterance by a human listener.

This paper subscribes to the view of semantics presented by Fodor (1978). Fodor makes a distinction between classical semantics and semantics as it is used by many psychologists and workers in artificial intelligence. Classical semantics is concerned with the relation between words and the "real world". It deals with the problem of what the referents of words are, and how the association between words and their meanings is defined and used. The point of view of artificial intelligence has been that this connection is mediated by some internal model of the "real world". There is a connection between natural language and this internal model, and a connection between this internal model and the outside world. In other words, this knowledge base represents or models the world referred to by the input sentences.

Semantic interpretation can be seen as a translation function from the surface sentence to a representation in this internal model. This definition requires that the range of possible semantic interpretations be defined by the human internal representation system and that any proposed representation system for semantic knowledge be psychologically relevant.

Evaluating Representation Systems

The second premise of this thesis is that the existence of a "right" method of representation suggests criteria for judging prospective knowledge representation systems. Previous AI attempts at developing knowledge representation systems have been of two varieties. One approach has been to take a constrained subpart of the "real world" and model it completely. This has been the approach taken by Green, et al (1963), with their baseball question answering system, Bobrow (1968) with his algebraic word problem solver, and most impressively by Winograd's blocks world robot (1972). The rationale for this approach is that it allows the relatively simple construction of a system which is both complete and robust (within its subdomain). Having a complete system allows the observation of the interaction of the major parts (syntax, semantics, reasoning etc.). Presumably the knowledge thus gained can be applied to more ambitious endeavors. Unfortunately, the extendability of the system as a whole is unproven and generally questionable.

A second approach has been to attempt to discover a basic set of principles governing knowledge representation which can model any domain, including that referred to by natural language. This has been the approach taken by Raphael (1968), Schank (1972), Martin (1977), and Minsky (1975). This approach has the obvious advantage of encompassing a larger set of problems. There is less chance that any results are an artifact of the domain under consideration. This approach also has the disadvantage of being more difficult. There is less guidance from the domain (or maybe too much guidance from the domain) in determining the design of the system. But a major problem is that it is difficult to appraise the success of any resulting

system.

Approaching the problem from an engineering point of view one could argue that there is only one criterion of success: How well does the resulting system perform? This is not a reasonable point of view, however, given a problem of this magnitude. Constructing a complete, robust knowledge representation system for the entire domain referred to by natural language is an inordinate amount of work to require before a theory can be judged. On the other hand, if only a small system is constructed and presented as evidence of the power of the system then it is impossible to determine whether the power is a function of the subdomain modeled, i.e. whether this is really an instance of the first approach, but with more grandiose claims. This problem is aggravated further by the fact that most approaches to a knowledge representation system do not constrain its power beyond that of a Turing machine. They present guidelines and suggested principles of organization but do not enforce them. This is not a deficit in and of itself but it does make it difficult to judge an incomplete system. With no prior constraints on the design, and with a limited set of situations to explain, solutions to the knowledge representation problem can be arbitrarily ad hoc. With no robust knowledge representation systems, one can always claim that one's kludges would fall out of a complete system.

So, for a general theory, if it is impractical to construct a complete system to test it, how should it be evaluated? It is the position of this paper that a legitimate approach is to look at specific phenomena that require interaction between better understood components of a full system (e.g. parsing, common sense reasoning) and the knowledge representation system. In other words, instead of constraining the domain of representation, examine specific

facets of the system which must be present regardless of the domain. For example, instead of building a system which can completely model a blocks world or the statistics of baseball games, one might look at an existing theory and see how it handles a specific problem which arises in the parsing of unrestricted natural language sentences. This is precisely what will be done in the remainder of this paper. This paper is going to examine an existing knowledge representation system, Minsky's frame theory, in light of the problem of determining the syntactic structure of modifiers in noun groups.

III. Syntax

The problem of parsing noun groups is a subpart of the more general parsing problem. This adds complexity to it in the following ways. First of all, the problem cannot be limited to simply nouns. The same problem is encountered when parsing adjective lists. For example, there are two possible syntactic interpretations of the phrase "light red wagon". Nouns and adjectives can appear together in a noun phrase with no obvious syntactic constraints, as the following examples demonstrate.

Nouns can modify adjectives: [[noun adj] noun]
the brick red house
the Vermont grown strawberries

Nouns can precede adjectives in a modifier list: [noun [adj noun]]
the New Year's Eve fancy dress ball (Jesperson, 1968, p.17)
Firestone steel belted radial

A second problem is possibility of relative clauses. Any noun phrase can be followed directly by another noun phrase which is the beginning of a relative clause. For example:

The men women like best use Aqua Velva.
The store mother shops at doesn't sell junk food.

This means that whenever a parser is parsing a noun phrase it can never be sure whether it is parsing one noun or two nouns concatenated together. The extent to which this problem can (or should) be solved is not clear, since even people cannot resolve this problem all of the time. Many constructions of this type cause garden paths in sentences. (footnote 1)

Finally, when expanding the problem, we also come across the general problem of ambiguity. In the general parsing problem, some members of a noun group under consideration may in fact not be nouns. Bever (1970, p. 323) gives the following examples.

The plastic pencil marks easily.
The plastic pencil marks were ugly.
The plastic rose fell.
The plastic rose and fell.

There have been no generally accepted syntactic solutions to the problem of parsing noun groups. For one thing, most syntactic analyses of noun phrases have dealt only with their production, without considering the problem of recovering the structure from the produced string. But mainly, there have been few attempts to explain either the modifier structure of a simple noun phrase (e.g. "light red wagon") or the structure of a compound nouns (e.g. "water meter cover"). See Downey (1977) for a summary of previous analyses of compound nouns.

Syntactic considerations may constrain the possible interpretations of a given string of nouns, but the problem does seem to require the use of semantics. It is difficult to imagine a syntactic explanation for the different

syntactic structures of the following two phrases.

water meter cover

plastic meter cover

The two phrases are identical except for their initial words. Both initial words are mass nouns denoting physical compounds. Yet the relationship among the words in the first phrase seems to be [[water meter] cover] while the relationship among the words in the second phrase would most likely be [plastic [meter cover]]. A parser capable of parsing these phrases correctly would somehow have to incorporate the fact that meters are more likely to be made of plastic than used to 'meter' them, but are more likely to 'meter' liquids than to be made of them. (This is necessary but of course not sufficient.) While one might try to incorporate this into a syntactic theory (and people have, e.g. (Lees, 1960)), any ostensibly syntactic solution will have to be ad hoc, based on the underlying semantics.

So, a key problem in parsing noun phrases is the criteria by which a parser can determine the plausibility of a given modifying relationship. There may be other considerations, but this one is essential. Therefore, any general purpose knowledge representation system should have mechanisms for implementing a solution to it. After laying some ground work in the next section, this paper will apply these premises to a particular knowledge representation system, frame theory.

III. Knowledge Representation

Knowledge Representation Systems

Up to this point, this paper has claimed that a solution to the parsing problem requires access to semantic knowledge, i.e. a knowledge base which represents or models the external world. A natural language parser must be able to appeal to this knowledge system when syntactic information is not enough to parse a sentence and this system must somehow supply the parser with the semantic information necessary to complete the parse. This semantic information should allow the parser to parse commonly understood natural language sentences in the same way (i.e. with the same interpretation) that fluent speakers parse these sentences. (We add the qualification that the sentences be "commonly understood" because people's semantic intuitions are much more divergent than their opinions on syntax. Since people often differ among themselves on the interpretation of sentences we cannot expect any knowledge system to predict the intuitions of all people at all times.)

The knowledge representation system under consideration, frame theory, falls into the broad area of semantic networks. The term "semantic network" has been used to describe almost any representation system where structure is imposed by 'nodes', which represent objects of some kind, and 'links' which represent relations between objects. This work is summarized very completely in Brachman (1978).

Brachman points out that the semantic significance assigned these nodes and links has varied widely, from simple implementational mechanisms (like lists) to the OWL system of Martin (1977) where the structure of the net is defined by the structure of English and the semantics of the net is identical to the semantics of English. He makes a case for five distinct levels of semantic

significance, and claims that each of these levels is important, in that each level should be able to be used to implement the level above it.

This paper will borrow two of his levels, what he calls the "epistemological" and "conceptual" levels. According to Brachman, "the epistemological level of "semantic network" permits the formal definition of knowledge-structuring primitives, rather than particular knowledge primitives (as in the Schank networks)" (1978, p. 35). The conceptual level is "... a network structure whose primitives are word-senses and case relations... in these conceptual level nets, links represent semantic or conceptual relationships. Networks at this level can be characterized as having small sets of language-independent conceptual elements (namely, primitive object- and action-types) and conceptually primitive relationships (i.e. "deep cases") out of which all expressible concepts can be constructed." (1978, pp. 32-33) Knowledge is represented by a conceptual level network but the ways in which it can be represented or structured are defined by the underlying epistemological level.

This enables the clarification of some of the points in the first section. The "world model" referred to there is a conceptual level semantic network. As Brachman noted, nodes in this network can correspond to "word-senses", or concepts. The concept [man], for example, is a node in this network which is distinguished by its contents and links to other nodes (and by its relationship to the word 'man' and the set of all men).

This world model is in turn based on, or implemented in, an epistemological level network. The epistemological network is not so much a tool as a building material. New concepts are added by constructing them out of this network, old concepts are altered by changing their implementation in it. The knowledge that is represented can vary from system to system or

person to person but the underlying implementation mechanism remains constant.

This implies that a solution to the problem of parsing noun groups should be done in terms of an epistemological level semantic net. Parsing noun groups is a fundamental ability common to all language users. It is not dependent on any specific knowledge contained in a world model, and is easily extended to new concepts added to the model. While a solution at the conceptual level is conceivable (see Gershman, 1977) it is not clear how it could be easily adapted to changes in the world model. (footnote 2)

Frame Theory

Frame theory is a combination of both an epistemological and conceptual level network. In addition to proposing the epistemological primitives of "frames" and "slots", Minsky suggests ways of organizing and manipulation them to solve problems in such areas as vision and natural language processing. Since it is the epistemological part which is relevant to the problem at hand, that part will be emphasized in the following discussion.

Frame theory is intended as "a partial theory of thinking" (Minsky, 1975, abstract). The theory is centered around objects, called "frames", described as follows.

> "Whenever one encounters a new situation (or makes a substantial change in one's viewpoint) he selects from memory a structure called a frame, a remembered framework to be adapted to fit reality by changing details as necessary.
>
> A frame is a data-structure for representing a stereotyped situation, like being in a certain kind of living room, or going to a child's birthday party. Attached to each frame are several kinds of information. Some of this

> information is about how to use the frame. Some is about what one can expect to happen next. Some is about what to do if these expectations are not confirmed.
>
> The "top levels" of a frame are fixed, and represent things that are always true about the supposed situation. The lower levels have many "slots" that must be filled by specific instances or data." (1975, abstract)

The key idea here is that of an amalgam of information in different forms (procedures, other frames, etc.) and functions united in its connection to a single concept. This information is structured through the use of slots. Slots are containers for information about frames. The contents of a slot is specified as being "data" or "instances". An important aspect of a slot is that, if not explicitly assigned, it is assumed to have a default value weakly bound to it.

In addition to this frame network there is a control mechanism which coordinates use of the frame system to achieve its goals. Minsky recognizes the need for a well defined control mechanism but does not go into detail about its possible structure.

As Minsky admits, this is an incomplete theory. It was proposed not as a complete theory but as a partial mechanism which "could help explain a number of phenomena of human intelligence" (1975, abstract). It will have to be further specified before it can be applied to the problem of parsing noun groups. Fortunately, the ideas of frame theory have been implemented in LISP by the language FRL (also, incidentally, by KRL (Bobrow and Winograd, 1977)). This implementation required a further specification of the problem which gave more structure to the idea of a frame.

FRL

FRL is an implementation of frame theory in LISP. Its purpose is to provide a convenient way of constructing a frame based knowledge system. As the manual states:

> The Frame Representation Language (FRL) is an experimental language written to explore the use of frames as a knowledge representation technique. The term "frame" as used in FRL was inspired by Minsky's [75] development of frame theory. FRL extends the traditional Property List representation scheme by allowing properties to have comments, defaults and constraints, to inherit information from abstract frames of the same type, and to have attached procedures triggered by adding or deleting values." (Roberts and Goldstein, 1977a, abstract)

Frames are implemented by FRL as collections of slots and are organized into a hierarchical classification system which enables the inheritance of properties. Each frame has one slot, the AKO (A Kind Of) slot, whose value is one or more other frames. Each frame is considered an "instance" of the frames in its AKO slot and inherits properties from these frames. Slots and the information they contain are inherited by instances of their containing frames. Other than the AKO slot (and its inverse, the INSTANCE slot) there are no restrictions on the number or type of slots. There are no other standardized slots in FRL. (footnote 3)

Slots are implemented as collections of facets. The addition of facets is intended to formalize some of the ideas presented in frame theory but not completely specified. It is intended that there be a small set of facet types which are used to structure the data contained by the slots in a uniform way. The current set of facets is:

VALUE - one or more value of the slot. There is no constraint on the form of a value.
DEFAULT - a value to use if none is explicitly assigned

IF-ADDED - function to be evaluated if a value is assigned to the slot
IF-REMOVED - function to be evaluated if a value is removed from the slot
IF-NEEDED - arbitrary LISP function which can be evaluated to determine a value of the slot, if needed.
REQUIRE - a requirement that must be true of any proposed value. Contents is a predicate to be applied to the proposed value. (Goldstein and Roberts, 1977b, pp. 13-16)

As the above quote indicates, FRL is intended primarily as a sophisticated data base. It provides built in functions for creating and destroying frames and for adding and retrieving values, taking advantage of the special facets and the inheritance hierarchy, but FRL says nothing about the possible form of a control mechanism. FRL is intended as an extension to LISP and the frame data base is to be used by an unspecified set of LISP procedures.

IV. The Problem, and a Solution

Another Look at the Problem

As noted above, there are no obvious syntactic constraints on the possible syntactic structure of a noun group. There also seems to be no obvious semantic constraints. First of all, it is possible to imagine a semantic relationship between any two random nouns. The relationship may be obscure or ad hoc, but given the appropriate context is perfectly understandable to people. For example, in a specific situation two nouns representing physical objects can be combined with the delimiting sense that the intended instance of the second object is the one in the physical proximity of the first. Downing (1977, p. 818) gives the example (taken from actual speech) "apple juice seat", the seat next to the glass of apple juice at the time the phrase was said. A

second point is that for many pairs of nouns more than one semantic relationship can be imagined between them.

So, while there are presumably at least some a priori constraints, this paper will assume that there are none. In other words, it will assume that the syntactic tree can have any branching structure.

This means that given a string of N nouns there is an a priori choice of (N-1)! possible structures. While all of these are perhaps conceivable, given a particular world model and context one of them is preferable to the others. The problem is then to use the world model to choose the appropriate structure.

Each of the prospective analyses can be defined by the modifying relations it imposes on the component nouns. For example, there are two prospective analyses for the phrase "water meter cover". One requires that "water" modify "meter" and "water meter" in turn modify "cover". The other requires that "meter" modify "cover" and "water" modify "meter cover". Each of these modifying relationships is plausible to some degree. The correct solution depends on the plausibility of each of these relations in the given world model. Therefore, to solve the problem we need to be able to characterize these relations in terms of a knowledge representation system.

The Problem in Terms of Frame Theory

The logical semantic correlate of a noun in frame theory is a frame (footnote 4). For example, presumably there is a frame for the concept [man], defined by its contents and its relations with other frames.

If nouns correspond to frames, then the relations among them must be

captured in the network imposed by the slots. The contents of a slot can be another frame. This connection expresses a relationship between the two frames.

The idea of a modifying relationship is represented straightforwardly in frame theory. The contents of a frame are not generally completely specified. The values of some slots are unknown, others are weakly bound defaults. The modification of this frame results in a more completely specified frame. Information from the modifying frame is used to "fill out" the modified frame. Note that this conveniently takes care of the extension from compound nouns to noun groups of three or more. Since the result of one frame modifying another is itself a frame, the resulting frame can modify or be modified by still another frame.

So, a solution to the parsing problem requires examining each of the modifying relationships entailed by each of the a priori possible solutions. For each of these modifying relationships, some measure is needed of the plausibility of the modifying frame further specifying the modified frame. This is as much of the solution as can be specified under the aegis of frame theory. Frame theory as presented by Minsky does not go into detail about the structure of slots or mechanisms for determining the structure or significance of a slot. Mechanisms could be proposed to solve this problem but since FRL goes into more detail in its implementation of frame theory, this discussion will continue with it.

The Problem in Terms of FRL

Everything developed so far can be extended easily to FRL since FRL takes away none of the structure proposed by frame theory. In fact by explicitly bringing out the idea of an AKO hierarchy it clarifies a point. The modified frame corresponding to a compound noun is presumably a new or preexisting frame which is an instance of the frame corresponding to the head noun. FRL also adds structure to the idea of a slot, and this structure can be used to further specify a solution to the problem.

What is needed is a way to characterize both the nature and plausibility of a modifying relationship. For simplicity, this paper will constrain the nature of the modifying relationship. It will assume that the modification is indicated by the modifying frame filling a slot in the modified frame. For example, with the phrase "water meter", the frame for the concept [meter] has (or inherits) a slot, call it the MEASURES slot, which has a requirement or preference that it be filled by a frame that is AKO the frame representing the concept [liquid]. The modified frame is AKO the [meter] frame, with the difference that its MEASURES slot has the [water] frame as a value. This general idea seems to characterize the majority of cases of noun noun modification. It isn't complete (footnote 5) but it will work as a first approximation and will serve to give some insight into the use of FRL.

Before continuing, an example is in order. The following is a representation of the two possible resolutions of the phrase "water meter cover". The following frames are presumed to exist in the knowledge base.

```
(FRAME "phys-obj")                (FRAME "solid")

(FRAME "liquid")                  (FRAME "water"
                                    AKO "liquid")

(FRAME "meter"                    (FRAME "cover"
```

```
AKO "phys-obj"                    COVERS "phys-obj"
MEASURES (AKO liquid))            MADE-OF "solid")
```

The frames on the left below correspond to the resolution [[water meter] cover]. The frames on the right correspond to the resolution [water [meter cover]]. Note the conflict between the contents of the MADE-OF slot of "water meter cover" and the contents of the MADE-OF slot of its superior frame, "cover". It is this conflict which indicates that the first solution is preferable to the second.

```
(FRAME "water meter"              (FRAME "meter cover"
  AKO "meter"                       AKO "cover"
  MEASURES "water")                 COVERS "meter")

(FRAME "water meter cover"        (FRAME "water meter cover"
  AKO "cover"                       AKO "cover"
  COVERS "water meter")             MADE-OF "water")
```

Solution 1 Solution 2

The problem can now be divided into two subproblems (although in an actual implementation they would probably be solved concurrently). First, generate the set of possible semantic analyses. This involves taking each possible syntactic analysis (it was previously assumed that there were (N-1)! of these for a string of N nouns) and, for each of the N-1 syntactic modifying relationships entailed, determining which slots in the modified frame could possibly be filled by the modifying frame. (e.g. for the syntactic analysis [water [meter cover]], determining how many ways "meter" can modify "cover" and how many ways "water" can modify "meter cover".) If each of the modifying relationships is semantically possible in at least one way then at least one semantic analysis can be generated from this syntactic analysis.

The second subproblem is, given the set of possible analyses, generate the

relative plausibility of each. The most plausible one should be the intended interpretation. Determining the plausibility of a semantic interpretation requires determining the plausibility of the modifying relationships entailed by it. The plausibility of a given modifying relationship depends on two things. First, how well does the modifying frame fit the slot in the modified frame? For example, meters can conceivably meter both plastic and water, yet the "water" frame somehow fits better into the MEASURES slot than the "plastic" frame. A second consideration is the plausibility of the slot being filled, the "relevance", in some sense, of the slot. For example, the phrase "water meter", meaning a meter that measures water is more plausible than the phrase "garden meter", meaning the meter located out in the garden. The MEASURES slot is more significant than the LOCATION slot in the "meter" frame.

The ideas are relatively straightforward. It would seem a simple matter to implement them in FRL. Unfortunately this is not the case. FRL does not provide the primitives to adequately implement them. It does provide some way of solving the first subproblem, that of determining what can fill a slot. Recognizing this need, it provides the REQUIRE facet. However, even this is somewhat ad hoc in that the contents of this slot is an arbitrary LISP predicate, evaluating only to true or false. There is no way for the system to determine how or why a particular decision was made.

But it is in the second subproblem that this lack of information causes serious difficulties. The second subproblem requires evaluating the plausibility of the proposed contents of a slot. There is no way this can be done with the existing primitives. Simple predicates cannot capture the fact that a slot can conceivably contain a wide variety of values, depending on the particular context.

As for the second part of the second subproblem, that of discriminating among slots, FRL does not even acknowledge the issue. FRL provides no built in mechanism for discriminating among slots, not even to the extent of distinguishing the AKO slot from the others. A frame is composed entirely of slots. This implies that there must be a wide variety of slots containing diverse kinds of information. Yet a procedure examining an unknown frame has no idea what is there, all of the data is a homogeneous mass.

Implications of the Solution (or Lack Thereof)

Implications for FRL

One might argue that this is being unduly harsh. After all, a solution to this problem could be implemented in FRL. For example, to judge the relative importance of slots, there could be a special slot containing a description (in a predetermined format) of each of the other slots. The name of this slot would be known to the controlling procedures and any information necessary to judge the relative plausibility of a given slot being filled could be contained in it. Given the appropriate information in the slot, along with the appropriate controlling LISP procedure to evaluate it, any conceivable solution could be implemented, including, most likely, the "right" one.

But of course this isn't right. This isn't an implementation in FRL, it is an implementation in LISP. The constructs of FRL are peripheral to the actual solution. It was argued earlier that parsing noun groups is an epistemological problem which should be solved in terms of the primitive constructs provided by an epistemological semantic net. Arbitrary LISP functions and S-lists are of

course in no sense FRL primitives.

This point is crucial. If FRL is being judged simply as an implementational mechanism for complex processes then this is by and large irrelevant. If that is the case, if it is simply an extended version of LISP, then is should be judged by the same criteria as other computer languages, e.g. by its simplicity, ease of programming, ease of representation, maintainability, etc. By these criteria, the only judgement this paper can pass on FRL is that it is slightly better than LISP as an implementation mechanism for a solution to the problem of parsing noun groups. It can say nothing about its adequacy for other problem areas.

However, FRL is presented as more than this. It is presented as an epistemological semantic network, if only in its name and connection with frame theory. On this basis, it should be expected to enable a reasonably straightforward solution to the problem of parsing noun groups, and its performance in this respect is a legitimate basis for criticism.

What can be concluded about FRL on this basis? First of all, it is incomplete. It does not specify a complete set of epistemological primitives. It makes up for this inadequacy by adding all of the power of LISP, and so is a mixture of frame theory and LISP. This is a mixture of two mechanisms with very different objectives. The object of frame theory is a psychologically relevant knowledge representation system while the object of LISP is a useful Turing machine. This causes two problems. First, it requires that structures that should be primitive in FRL must be implemented in LISP by the user. Also, the inclusion of LISP allows the misuse or circumvention of the primitives that FRL does provide. An example of this is in the proposed solution at the beginning of this section. FRL is incomplete in that it does not

differentiate among slots. The proposed solution involved both the adding of primitive function through LISP and the misuse of the concept of a slot.

A second conclusion that can be made is that, while FRL adds needed structure to frame theory, it is not clear that this structure is appropriate. The addition of a REQUIRE facet filled explicitly by a LISP predicate appears to be a step in the wrong direction. Rather that opaque LISP code, the parsing problem requires a description of the possible contents of the slot that can be manipulated in a uniform way.

One possible approach might be to represent all data as frames. Then the REQUIRE facet could contain a "model" frame characterizing the possible contents of the slot. This "model" frame would be a partial frame containing those parts of a frame which must be present in any prospective frame which fills the slot. This handles some cases easily. For example, the REQUIRE facet for the LOCATION slot would simply be a model frame with only the AKO slot filled out, being AKO an abstract "place" frame. Other cases are not so simple. It is not immediately obvious what the "volt" and "water" frame (or "water" and "postage") would have in common that would indicate their preferred positions as possible instantiations of the MEASURES slot of the "meter" frame. It could turn out that what is required is something with all of the descriptive power of LISP, but with a structure which can be readily analyzed by the controlling processes.

Implications for Frame Theory

What has all of this indicated about frame theory? First of all, frame theory certainly allows the primitives necessary for a solution. And, as noted,

the problem maps nicely into the primitives specified. The ideas of frames further specifying other frames nicely characterizes the modifying relationship.

However, also as noted, frame theory is definitely incomplete, and is not specified sufficiently to construct a complete solution to the problem of parsing noun groups. While certainly suggesting the differentiating of slots, it does not specify how this differentiation might be expressed, or how the frame as a whole might be structured. Another deficiency is the lack of detail in the description of a slot. What exactly are slots used for and what semantic significance is assigned this use? What can slots contain and how is this indicated? While the overall sketch of the system is eminently reasonable, it seems necessary to fill in more of the detail before the theory can be practically applied.

There is one aspect of frame theory which must be brought out. This is the control mechanism. FRL demonstrates the problem with specifying a data representation system without specifying the processes which manipulate the data. If the control mechanism is unconstrained then the theory is both incomplete and open to abuse. Since programs and data are in a sense equivalent, a complete theory must describe both. If the program is not specified then function no included in the data base is unspecified and function supposedly included in the data base can in fact be implemented in the program. Minsky recognizes the need for a control mechanism, this thesis emphasizes it.

VI. Summary

The intention of this thesis has been to demonstrate a legitimate and effective approach to the problem of representing and utilizing semantic knowledge. It has argued that a proposed semantic knowledge base should mimic the human mind, and that there is an especially appropriate method of evaluating a knowledge representation system claiming psychological relevance.

A knowledge representation system can be thought of as existing on two levels. On one level is the underlying epistemological primitives. This is the "building material", or representation system out of which the knowledge is structured, and which defines the range of possible knowledge structures. The second level is the knowledge system resulting from arranging these primitives and assigning semantic significance to them (in the classical sense). The first level is static, the second is dynamic, changing as more information is assimilated by the system. The assumption here is that people share a common epistemological base which they use to shape there own individual knowledge bases.

The use of uniform epistemological primitives simplifies the controlling procedures which make use of the knowledge represented. The use of a particular epistemological system will make various operations more or less viable. This presents a method of evaluating prospective artificial representation systems.

A prospective knowledge representation system should reflect the same relative efficiencies and inefficiencies (as well as possibilities and impossibilities) as are present in the human representation system. The degree to which it does is a measure of its psychological relevance, and, in turn, of its absolute relevance as a general purpose representation system for semantic knowledge.

This approach was demonstrated by analyzing the performance of a proposed knowledge representation system, frame theory, with the problem of determining the semantic relationships among the components of noun groups. It was argued that frame theory was in fact proposing epistemological primitives and that parsing noun groups was a problem that must be solved primarily in term of an epistemological system.

The result of this analysis was that frame theory as specified by Minsky was found to be insufficiently specified to solve this problem. The primitives of frame theory do not impose the necessary structure on the information represented. The implementation of frame theory in FRL was found to provide more structure in its primitives, but was deficient in two ways. First, it did not provide a complete set of primitives, a complete epistemological level, but rather implicitly assumed the implementation of some primitives in LISP. The second deficiency was that some of the added structure it provided, while supposedly relevant to the problem at hand, proved ineffective and essentially useless in that it would have to be duplicated in a different fashion in order to be used in a solution.

The final section of the paper brought up similar problems that are presumably solved in terms of epistemological primitives. It was claimed that any psychologically relevant knowledge representation system will have primitives that enable the solution of these problems in a natural way.

Footnotes

1) Marcus' parser (1978) solves this problem in a way that reflects the performance of people on these sentences, but his solution presupposes the separation of the two noun groups. There is still the problem of, given the linear string of words, how can one determine that there are in fact two nouns present?

2) Of course, any system presumably has methods for adding connections between particular nouns which are commonly found together. For example, after hearing the phrase "peanut butter" several times, people probably generate a connection corresponding to the relation in their internal knowledge base which simplifies future parsing. This thesis, however, is concerned primarily with how phrases with no a priori connections are parsed, e.g. how one parses "peanut butter" the first time it is heard. This issue is related to the phenomena of lexicalization, where some compound nouns acquire standardized meanings, e.g. "bellboy" or "lipstick".

3) Note that the implementation of the AKO relation as a slot value is an unfortunate mixing of the epistemological and conceptual levels. The idea of a slot and the idea of an AKO hierarchy are epistemological level primitives. However, the particular interpretation assigned to the contents of a slot in a given knowledge representation system (e.g. MOTHER-OF, LEFT-OF) is a conceptual primitive. The AKO slot is, in fact, not really a slot at all.

4) There could be frames for other words to. In fact, a reasonable view is that the set of frames corresponds roughly (but certainly not exactly) to the set of open class words in a person's vocabulary, on the hunch that words are words for a reason. This would enable the extension of this solution to the case of adjectives.

5) This assumption is obviously incomplete in cases of metaphoric modifiers such as "starfish" or "skijump nose" where it isn't the modifying noun which is further specifying the head noun but rather some distinguishing characteristic of the modifying noun. It also has problems with reversible phrases like "male caucasian" and "caucasian male" which presumably are resolved into the same frame. Finally, it is totally incapable of the general purpose reasoning required to resolve (incorrectly, but humanly) the phrase "ten gallon hat".

Bibliography

Bever, T. G. 1970. "The Cognitive Basic for Linguistic Structures." In Cognition and the Development of Language. John R. Hayes, ed. John Wiley and Sons, Inc.

Bobrow, Daniel G. 1968. "Natural Language Input." In Semantic Information Processing. Marvin Minsky, ed. Cambridge, MA: The M.I.T. Press, pp. 146-226.

Bobrow, Daniel G., and Winograd, Terry. 1977. "An Overview of KRL, a Knowledge Representation Language." Cognitive Science, Vol. 1, No. 1, January, 1977, pp. 3-46.

Brachman, Ronald J. 1978. "On the Epistemological Status of Semantic Networks." BBN Report No. 3807. Cambridge, MA: Bolt Beranek and Newman Inc., April 1978.

Clark, Eve W., and Clark, Herbert H. 1978. "When Nouns Surface as Verbs." unpublished ms.

Downey, Pamela. 1977. "On the Creation and Use of English Compound Nouns." Language, Vol. 53, No. 4, pp. 810-842.

Fodor, J. A. 1978. "Tom Swift and His Procedural Grandmother." Unpublished ms. M.I.T.

Gershman, Anatole V. 1977. "Conceptual Analysis of Noun Groups in English." In Proceedings of the Fifth International Joint Conference on Artificial Intelligence, Cambridge, MA, August, 1977, pp. 132-138.

Green, Bert F. 1963. "Baseball: An Automatic Question Answerer." In Computers and Thought. Edward A. Feigenbaum and Julian Feldman, eds. New York: McGraw-Hill Book Company, pp. 207-216.

Goldstein, Ira P., and Roberts, R. Bruce. 1977. "Nudge, a Knowledge-based Scheduling Program." In Proceedings of the Fifth International Joint Conference on Artificial Intelligence, Cambridge, MA, August, 1977, pp. 257-263.

Jesperson, Otto. 1969. Analytic Syntax. New York: Holt, Rinehart and Winston, Inc.

Lees, R. B. 1960. "The Grammar of English Nominalizations." International Journal of American Linguistics, Vol. 26, No. 3, Part II.

Marcus, Mitchell P. 1978. "A Theory of Syntactic Recognition for Natural Language." unpublished PhD thesis, M.I.T.

Martin, William A. 1977. "OWL". In Proceedings of the Fifth International Joint Conference on Artificial Intelligence, Cambridge, MA, August, 1977, pp. 985-987.

Minsky, Marvin. 1975. "A Framework for Representing Knowledge." In The Psychology of Computer Vision. Patrick H. Winston, ed. New York: McGraw-Hill Book Company, pp. 211-277.

Raphael, Bertram. 1968. "Semantic Information Retrieval." In Semantic Information Processing. Marvin Minsky, ed. Cambridge, MA: The M.I.T. Press, pp. 33-145.

Roberts, R. Bruce, and Goldstein, Ira P. 1977a. The FRL Primer. A.I. Memo No. 408. Cambridge, MA: Artificial Intelligence Laboratory, M.I.T., April, 1977.

Roberts, R. Bruce, and Goldstein, Ira P. 1977b. The FRL Manual. A.I. Memo No. 409. Cambridge, MA: Artificial Intelligence Laboratory, M.I.T.,

September 1977.

Schank, Roger C. 1972. "Conceptual dependency: A Theory of Natural Language Understanding." Cognitive Psychology, 3, pp. 552-631.

Winograd, Terry. 1972. Understanding Natural Language, PhD thesis, Academic Press, 1972.

Winston, Patrick H. 1975. "Learning Structural Descriptions from Examples." In The Psychology of Computer Vision. Patrick H. Winston, ed. New York: McGraw-Hill Book Company, pp. 157-209.